



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

IDENTIFIKACE OBJEKTŮ V OBRAZE

THE IDENTIFICATION OF OBJECTS IN AN IMAGE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Filip Štěpán

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Libor Boleček, Ph.D.

BRNO 2016



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Filip Štěpán

Ročník: 3

ID: 147372

Akademický rok: 2015/2016

NÁZEV TÉMATU:

Identifikace objektů v obraze

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s identifikací objektů v obraze pomocí rozhodovacích stromů. Volba aplikační úlohy identifikace objektů v obraze (např. dopravní značky, předměty na běžícím pásu ve výrobě, kontrola přítomnosti součástí složitějšího celku).

Návrh algoritmu rozhodovacího stromu pro zvolenou aplikační úlohu. Vytvoření softwaru s uživatelským rozhraním v prostředí MATLAB případně jiném. Experimentální otestování navrženého rozhodovacího stromu a vytvořené aplikace.

DOPORUČENÁ LITERATURA:

[1] GONZALES, C.G., WOODS, R.E., EDDINS, S.L. Digital Image Proccessing Using MATLAB, Gatesmark publishings, 2009.

[2] ZAVALINA, V. Identifikace objektů v obraze. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2013. 66 s., Diplomová práce. Vedoucí práce: ing. Libor Boleček.

Termín zadání: 8.2.2016

Termín odevzdání: 26.5.2016

Vedoucí práce: Ing. Libor Boleček, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Tomáš Kratochvíl, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem práce je vytvoření algoritmu pro identifikaci objektu v obraze. Výsledný program bude schopen zpracovat reálná obrazová data. Z tohoto důvodu je důraz kladen zejména na složitost prostředí, ve kterém se objekty nalézají, spolehlivou segmentaci a kvalitní identifikaci objektu. Práce je zaměřena na detekci a identifikaci svislého dopravního značení v obraze, nacházejícího se podél silničních komunikací na území České republiky.

KLÍČOVÁ SLOVA

Detekce, identifikace, rozpoznání, rozhodovací stromy, dopravní značení, HSV, segmentace

ABSTRACT

The aim of this paper is the creation of an algorithm for identification of objects in an image. The resulting programme will be able to process real visual data. This is the reason why the emphasis is put mainly on the complexity of the environment in which the objects are found, reliable segmentation and quality of the object identification. The paper is further focused on the detection and the identification of vertical traffic signs in an image that are found alongside road communication on the territory of the Czech Republic.

KEYWORDS

Detection, identification, recognition, decision trees, traffic signs, HSV, segmentation

ŠTĚPÁN, F. *Identifikace objektů v obraze*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 69 s. Vedoucí bakalářské práce Ing. Libor Boleček, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Identifikace objektů v obraze“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Tímto bych rád poděkoval vedoucímu bakalářské práce panu Ing. Liboru Bolečkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	1
1 Metody pro rozpoznání dopravního značení	2
2 Obrazová data	3
2.1 Vybrané barevné modely	3
2.1.1 Prostor RGB	4
2.1.2 Prostor CMYK	5
2.1.3 Prostor HSV	6
2.1.4 Prostor YCbCr	7
2.2 Šedotónový obraz	8
2.3 Binární obraz	9
3 Rozhodovací stromy	10
3.1 Kteriální statistika	10
3.2 Algoritmy růstu stromu	12
3.2.1 Učení s učitelem	13
3.2.2 Algoritmus CART	13
3.2.3 Algoritmus ID3	13
3.2.4 Algoritmus CHAID	14
3.3 Prořezání stromu	14
4 Zpracování obrazu	16
4.1 Jasová transformace	16
4.1.1 Lineární zvýraznění kontrastu scény	16
4.1.2 Ekvalizace histogramu	18
4.2 Geometrická transformace	19
4.2.1 Změna měřítka	19
4.2.2 Rotace	21
4.2.3 Zkosení	21
4.2.4 Složené transformace	21
4.3 Detekce hran	23
4.3.1 Hranové detektory aproximující první derivaci obrazové funkce	23
4.4 Segmentace	24
4.4.1 Segmentace prahováním	25
4.5 Matematická morfologie	25
4.5.1 Operátory MM	25

5	Popis a identifikace objektů	28
5.1	Identifikace oblastí	28
5.1.1	Dvouprůchodový algoritmus	28
5.2	Radiometrické deskriptory	29
5.2.1	Deskriptory založené na regionech	29
5.3	Houghova transformace	30
6	Techniky rozpoznání	33
6.1	Template matching	33
6.1.1	Korelace	33
6.1.2	Křížová korelace	34
6.2	Algoritmus SURF	34
6.2.1	Detekce významných bodů	34
6.2.2	Scale-space	35
6.2.3	Přiřazení orientace významným bodům	36
7	Implementace	37
7.1	Rozhodovací strom	38
7.2	Předzpracování	39
7.3	Segmentace	40
7.4	Popis objektů	42
7.5	Klasifikace radiometrickými deskriptory	42
7.6	Klasifikace Houghovou transformací	44
7.7	Rozpoznání	45
7.7.1	Template matching	46
7.7.2	SURF	48
7.8	Databáze značek	49
8	Výsledky algoritmu	51
8.1	Statistické vyhodnocení úspěšnosti	56
9	Uživatelské prostředí	61
	Závěr	63
	Literatura	65
	Seznam symbolů, veličin a zkratk	68
A	Obsah přiloženého CD	69

SEZNAM OBRÁZKŮ

2.1	Chromatický diagram CIE, znázornění gamutů (Převzato z [5]). . . .	4
2.2	RGB model. [2]	5
2.3	HSV model jako kužel. [6]	7
4.1	Příklad roztažení a ekvalizace histogramu.	17
4.2	Příklad změny měřítka objektu podle velikosti vzoru.	20
4.3	Příklad geometrických transformací.	22
5.1	Příklad Houghovy transformace.	31
6.1	Aproximace druhé derivace Gaussovy funkce (zleva doprava) podle x, y a dle xy. Obrázek je inspirován [24].	35
6.2	Filtrovací jádro metody SURF. Obrázek je inspirován [24].	36
7.1	Schéma algoritmu.	37
7.2	Rozhodovací strom.	38
7.3	Vývojový diagram procesu segmentace s ohledem na barvu.	40
7.4	Vývojový diagram procesu segmentace s ohledem na velikost.	41
7.5	Vývojový diagram procesu detekce tvaru.	43
7.6	Vývojový diagram algoritmu <i>template.m</i>	47
7.7	Ukázka výsledných korespondujících bodů metody SURF.	49
8.1	Ukázka správné identifikace pomocí metody SURF.	51
8.2	Ukázka segmentace předchozí scény.	52
8.3	Ukázka správné identifikace pomocí metody SURF.	52
8.4	Ukázka správné identifikace pomocí metody template matching. . . .	53
8.5	Ukázka správné identifikace pomocí metody template matching. . . .	53
8.6	Ukázka správné detekce pomocí Houghovy transformace.	54
8.7	Ukázka nulové detekce zapříčiněné nepříznivými světelnými podmín- kami.	54
8.8	Ukázka segmentace předchozí scény.	55
8.9	Detekce radiometrickými deskriptory.	58
8.10	Detekce Houghovou transformací.	58
8.11	Detekce radiometrickými deskriptory.	59
9.1	Uživatelské prostředí po spuštění.	61
9.2	Uživatelské prostředí po dokončení identifikace.	61

SEZNAM TABULEK

8.1	Tabulka zkratek možných kombinací metod.	56
8.2	Tabulka úspěšnosti identifikace metodou template matching.	57
8.3	Tabulka úspěšnosti identifikace metodou SURF.	59

ÚVOD

Detekce objektu v obraze je v dnešní době stále více využívána v různých odvětvích. Ať už se jedná o oblast průmyslu, kde se detekce využívá například k rozpoznání vady určité součástky, dopravní sektor, kde je detekce využita například k rozpoznání SPZ vozidel a jiné, například různé aplikace k rozpoznání obličeje nebo emocí (detekce úsměvu). Tato práce se zabývá problematikou detekce a identifikace svislých dopravních značek, nacházejících se podél dopravních komunikací na území České republiky.

Na toto téma již existuje celá řada úspěšných metod, které se dnes využívají například ve vozidlech pro autonomní řízení. Tato práce se bude zabývat metodou zahrnujícího barevnou segmentaci a detekci objektů dle předem známých tvarů. Následná identifikace bude pak provedena porovnáváním objektu s šablonou nebo pomocí srovnání významných bodů vzoru a šablony, získaných metodou SURF.

První část této práce se věnuje teoretickému rozboru problematiky detekce objektu v obraze. Zde jsou popsány vybrané barevné prostory, algoritmy rozhodovacích stromů a metody předzpracování obrazu, spolu s metodami detekce a identifikace objektů.

Ve druhé části je pak kladen důraz na využití znalostí získaných při studiu teorie a jejich použití v aplikaci. Do této problematiky spadá zejména úloha porozumění scéně, segmentace a identifikace segmentovaných oblastí. Výsledná aplikace bude navržena prostřednictvím programovacího prostředí Matlab, tohoto softwaru je využito zejména kvůli jeho jednoduchosti a možnosti využití celé škály funkcí různých toolboxů. Na aplikaci není stanoven žádný požadavek týkající se rychlosti či přesnosti, proto se nabízí možnost provést několik testování s aplikací, lišící se v určitých metodách a jejich následného porovnání mezi sebou. Zde nás bude zajímat zejména rychlost celého procesu detekce a identifikace a také přesnost.

1 METODY PRO ROZPOZNÁNÍ DOPRAVNÍHO ZNAČENÍ

V dnešní době jsou k těmto účelům stále více využívány tzv. evoluční algoritmy, například umělé neuronové sítě, které značně urychlují procesy detekce, třídění a identifikace hledaných objektů v obraze. Dalším možným přístupem spadajícím k evolučním algoritmům jsou pak algoritmy genetické. Procesy založené na neuronových sítích jsou velice blízké procesům biologických neuronů. Neuronové sítě však obvykle vyžadují poněkud rozsáhlý proces trénování.

Možných přístupů k rozpoznání dopravního značení je skutečně mnoho a ve většině případů jsou určeny hlavně požadavky na časovou náročnost procesu rozpoznání daného objektu v obraze. Například v diplomové práci [1] autor používá modifikace kaskádního detektoru Viola-Jones implementovaného v prostředí OpenCV. Výhodou této metody je její vysoká rychlost, mezi nevýhody patří zejména nutnost učení algoritmu na množinách pozitivních a rozsáhlých množinách negativních tréninkových vzorů. Autor práce uvádí úspěšnost 79%. Časová náročnost celého procesu je pak uváděna 400 ms.

Ve článku [2] autor popisuje metodu rozpoznání pouze červených dopravních značek, která je založena na barevné segmentaci v HSV prostoru. Ke klasifikaci jsou využity radiometrické deskriptory v kombinaci s Houghovou transformací a proces identifikace je zajištěn pomocí několika neuronových sítí. Autor dále uvádí úspěšnost této metody testované na celkem 200 snímcích 90,8% a časovou náročnost celého algoritmu 2 až 3 vteřiny (algoritmus je implementován v programu Matlab).

V dalším článku [3] se autoři zabývali pouze detekcí LED značení. Toto značení je čtvercového tvaru. Za účelem detekce tohoto druhu značení používají autoři tzv. *rectangular Hough transform*. Autoři uvádí úspěšnost detekce 85.37% při testování na 87 snímcích.

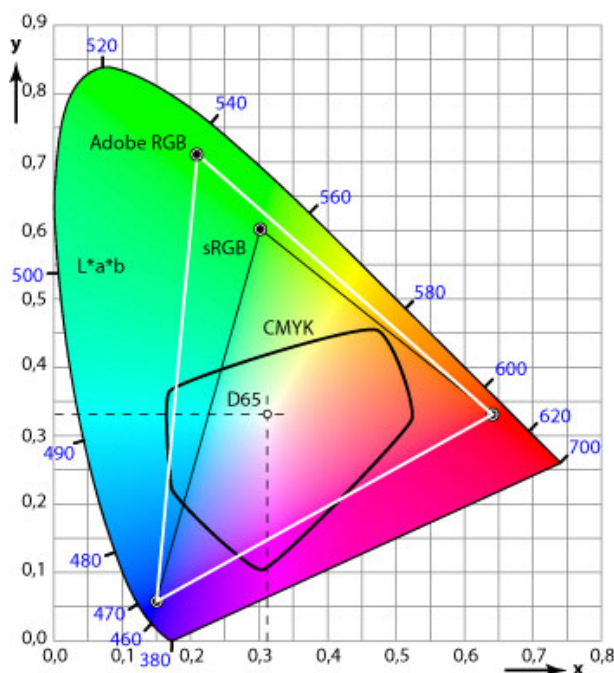
2 OBRAZOVÁ DATA

Lidské oko vnímá spojitě spektrum cca. od 380 do 720 nm. Člověk je tak schopen rozlišit zhruba 400 000 různých barev a jejich odstínů [4]. Avšak ani nejmodernější fotoaparáty nedokáží přesně napodobit celé barevné spektrum, které vnímáme. Důkazem toho jsou barevné modely, jež se snaží co nejlépe se přiblížit realitě snímané scény. Každý barevný model je využíván různými zařízeními. Od RGB modelu, využívaného právě digitálními fotoaparáty a takřka všemi druhy obrazovek přes CMYK model, používaný pro tisk, až po model HSV, který je našemu chápání barev nejbližší z uvedených. Tyto modely budou dále rozebrány v následujících podkapitolách.

2.1 Vybrané barevné modely

V této sekci je čerpáno z [5]. Barevných modelů, se kterými se můžeme setkat, je skutečně mnoho. My zde popíšeme pouze modely RGB, CMY(K), YCbCr a model HSV hlavně z důvodu, že tyto modely budou následně využívány pro práci se snímky (segmentaci) v navrženém programu. Model CMY(K) byl vybrán k popisu výhradně pro jeho vzájemnou souvislost s prostorem RGB.

Každý barevný model disponuje tzv. gamutem. To je rozsah všech barev (sytost), které je konkrétní model schopen reprodukovat. Nejlépe se gamut zobrazuje v chromatickém diagramu CIE, viz obr. 2.1, ve kterém jsou obsaženy všechny spektrální barvy, které je schopen člověk vnímat. Každý barevný model vyznačený v tomto diagramu zde zaujímá různě velkou plochu, liší se i tvarem a pozicí. Nastává tedy problém při převodu barev mezi dvěma různými modely, kdy některé barvy např. z RGB modelu nelze v CMY(K) modelu zastoupit a naopak. Tyto barvy se pak musí nahradit barvou nejpodobnější, která již v gamutu požadovaného prostoru leží. Tomuto procesu se říká komprese gamutu. Konkrétní rozsahy gamutu jednotlivých modelů jsou znázorněny na obrázku 2.1.

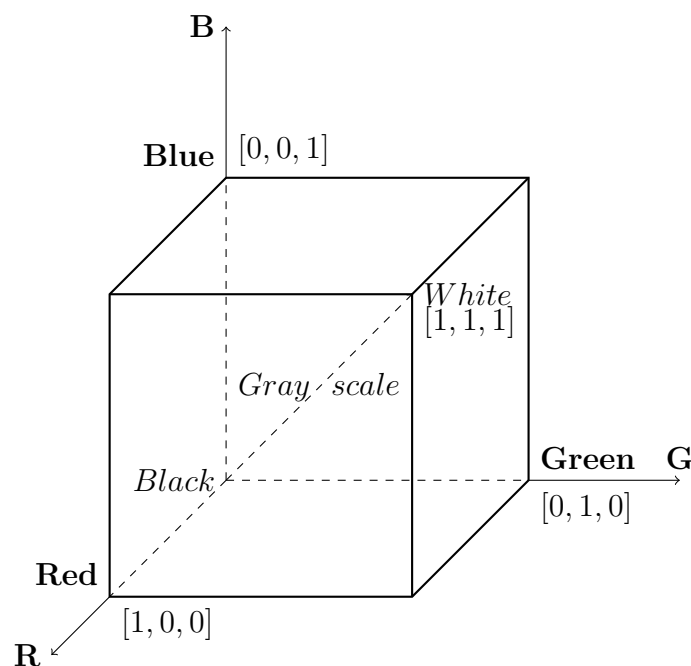


Obr. 2.1: Chromatický diagram CIE, znázornění gamutů (Převzato z [5]).

2.1.1 Prostor RGB

Teorie převzata z [6] a [7]. Barevný model RGB je jedním z nejběžněji používaných modelů pro digitální obrazy, využívají jej fotoaparáty, obrazovky, televize a projektory. RGB model se skládá ze tří základních barev: červené (R, red), zelené (G, green) a modré (B, blue). Tyto základní barvy můžeme dále kombinovat a vytvářet tak další odstíny. Tento model je založen na tzv. **aditivním** míšení barev. To znamená, že čím více barev smícháme, tím bude výsledná barva světlejší. Lze jej názorně zobrazit jako jednotkovou krychli umístěnou na osách označených R, G a B. Počátek souřadnic, tedy bod $[0, 0, 0]$, zde odpovídá černé barvě, bod $[1, 1, 1]$ pak značí bílou. Plné barvy se nacházejí na osách v bodech 1 pro každou z os, tzn. že například plná červená se bude nacházet na ose r v bodě 1, tj. bod $[1, 0, 0]$ v prostoru. Tyto barvy nazýváme primárními. Ostatní vrcholy krychle tvoří pak barvy doplňkové, označované jako sekundární. Takto popsáný prostor pak můžeme vidět na obrázku 2.2.

RGB obraz je trojrozměrné pole, tedy tři dvourozměrná pole charakterizující jeden z kanálů. Každých 8 bitů pro každou ze složek, tedy rozsah 0-255, pak charakterizuje výsledné 24 bitové vyjádření obrazu tzv. *true color*.



Obr. 2.2: RGB model. [2]

2.1.2 Prostor CMYK

V této sekci je čerpáno z [5], [6], [7]. Oproti RGB prostoru je zde míšení barev řešeno **subtraktivně**, čím více barev smícháme, tím tmavší je výsledná barva. CMY prostor je složen ze tří základních barev: modrozelená (C, cyan), fialová (M, magenta) a žlutá (Y, yellow). Tento model byl navržen a používá se výhradně v tiskových procesech. Tiskový pigment nesmí být dokonale krycí, aby bylo možno pigmenty mísit. Smícháním všech barevných pigmentů tedy nevznikne černá, jak ji známe, ale tzv. špinavě hnědá, navíc by kvůli černé barvě takto vytvořené bylo spotřebováno příliš mnoho toneru, z toho důvodu byla ještě přidána černá barva, viz model CMYK, kde K značí černou (K, black).

Tento prostor lze stejně jako předchozí popsat jednotkovou krychlí, avšak vrcholy černé a bílé jsou vzájemně prohozeny, jelikož subtraktivní součet všech barev v modelu má za následek vytvoření černé barvy (zde vrchol $[1, 1, 1]$). Již z představy této krychle vyplývá, že převod mezi RGB modelem, viz Obr. 2.2 a CMY modelem bude velice jednoduchý:

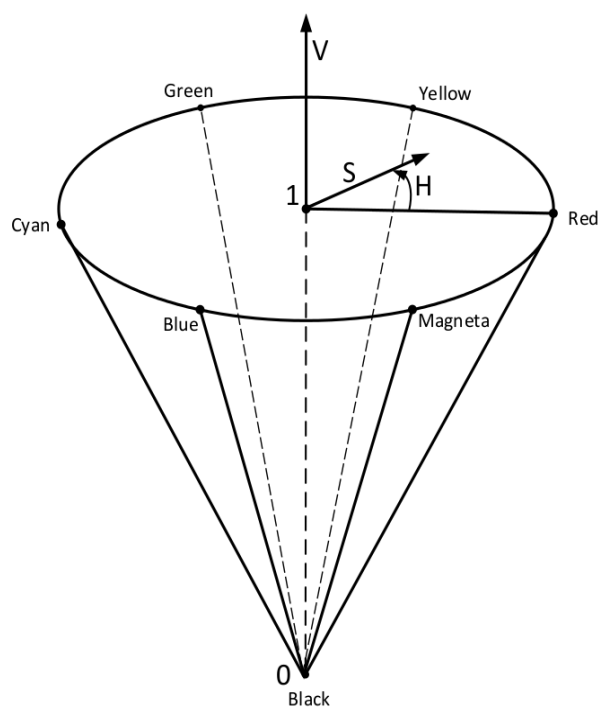
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.1)$$

Převod mezi těmito soustavami však není dokonalý a občas nelze jednu konkrétní barvu zobrazit přesně v druhém modelu. Výsledný obraz z tiskárny tedy nebude vypadat úplně stejně, jak jej vidíme na monitoru. Některé barvy vytvořené v RGB modelu nelze v CMY modelu napodobit, a tak se nahrazují nejbližší možnou barvou (komprese gamutu).

2.1.3 Prostor HSV

Čerpáno z [6] a [7]. Tento barevný model je našemu chápání barev nejbližší z uvedených. Model HSV, někdy označovaný jako HSB, se skládá taktéž ze tří složek, avšak na rozdíl od RGB nebo CMY modelu je informace o barvě obsažena hned v první složce zvané odstín (H, hue). Zde jsou konkrétní odstíny barev vyjádřeny úhlem od 0° do 360° . Konkrétní úhel pak tedy reprezentuje převládající spektrální barvu. Další složkou je sytost (S, saturation), která se mění od 0 do 1 a udává čistotu dané barvy. Poslední složkou je jasová hodnota (V, value), pohybuje se taktéž v rozmezí hodnot 0 až 1 a označuje množství bílého světla. Nespornou výhodou pro práci v tomto modelu je, na rozdíl od předchozích modelů, možnost oddělit jas a sytost a pracovat s nimi bez ohledu na barevný tón.

V Matlabu užíváme k převodu mezi soustavami RGB a HSV příkazu *rgb2hsv* respektive *hsv2rgb*. Jak již bylo zmíněno, nespornou výhodou tohoto modelu je samostatná matice jasu, se kterou můžeme pracovat a zároveň neovlivňovat zbylé matice pro barevný tón a sytost barvy. Z toho důvodu je model HSV výhodnější pro práci s barevnými obrázky a je proto využit v této práci.



Obr. 2.3: HSV model jako kužel. [6]

2.1.4 Prostor YCbCr

V této sekci je čerpáno z [5]. Zkráceně YCC není absolutním modelem, používá se pro vyjádření jinak kódovaného RGB signálu. Jeho výhodou je opět oddělený prostor jasu Y od ostatních Cb modrého a Cr červeného chrominančního komponentu. Model byl doporučen institucí CCIR pod označením 601 a využíván v barevných televizních soustavách, kde bylo potřeba uchovávat zvlášť informace o jasů kvůli (černobílým) televizorům. Používá se nejčastěji u videa či digitálních fotografií (zejména ztrátová komprese JPEG). Záměr použití vychází z citlivosti lidského oka na jas a barvy. Model umožňuje při komprimaci snížit hodnoty chrominančních složek (podvzorkování) a zároveň zachovat datovou velikost jasové složky. Poté následují další kompresní kroky, které však přesahují rámec této práce. Ze samotné definice prostoru vyplývá, že převod mezi modelem RGB a YCbCr bude následovný:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.2)$$

2.2 Šedotónový obraz

Čerpáno z [6]. Reprezentace šedotónového obrazu je znázorněna na obrázku 2.2 spojnici vrcholů pro bílou a černou barvu. Tedy je to právě úhlopříčka v krychli RGB spojující vrchol $[0, 0, 0]$ – černá barva a vrchol $[1, 1, 1]$ – bílá barva. Na této úhlopříčce jsou hodnoty všech barev stejné, a proto není potřeba 24 bitového vyjádření. Pro stupně šedi nám postačí 8 bitů. Stupně šedi se zde udávají stejně jako v předchozím případě, a sice jako hodnoty od 0–255. Abychom z RGB modelu dostali pouze šedotónový obraz, je zapotřebí provést určité matematické operace, které nám z barevného obrazu vytvoří obraz jasový. Možností je použít tzv. prosté průměrování, kde jsou koeficienty násobení každé barvy rovny jedné třetině, to si lze představit právě jako úhlopříčku v RGB krychli 2.2. Nebo převod používaný funkcí *rgb2gray*, který představuje složka Y z předchozí rovnice 2.1.4 tzv. *nonlinear luma*. Další možností je aplikace převodu definovaného standardem CIE:

$$Y = 0,2126R + 0,7152G + 0,0722B. \quad (2.3)$$

Koeficienty, které násobí každou barevnou složku, vycházejí ze známé citlivosti lidského oka (čípků) na tyto základní barvy.

Pokud bychom chtěli s obrazem pracovat a dále jej upravovat, museli bychom výsledný jasový snímek složit zpět do matice RGB. Adekvátní řešení je využít jiný barevný model, například HSV model, který již obsahuje jasové informace v jedné ze svých složek. Nabízí se tedy hned několik možností pro úpravu jasu, které jsou dále popsány v kapitole zabývající se předzpracováním obrazu.

2.3 Binární obraz

Teorie převzata z [6]. Binární obraz je kódován jako 2D pole. Používá 1 bit na pixel, kde 0 většinou značí černý pixel a hodnota 1 bílý. Hlavní výhodou tohoto znázornění scény je jeho malá velikost. Obvykle se využívá pro obrazy s jednoduchou grafikou, texty nebo kresby.

V Matlabu se používá pro vyjádření pro výsledný obraz hranových filtrů, některé funkce image processingu jej přímo vyžadují. Binární obraz nejlépe vytvoříme z šedotónového obrazu prahováním dle určitého prahu, tj. zvolíme si hodnotu stupně šedi, nad kterou budou všechny pixely považovány za bílé, a bude jim přiřazena hodnota 1, ostatní pixely budou považovány logicky za černou barvu a bude jim přiřazena hodnota 0. V Matlabu funkce *logical*, *bwlabel*, *bwmorph* apod.

3 ROZHODOVACÍ STROMY

Teoretické znalosti popsané v této kapitole byly čerpány zejména z [8] a [9]. Rozhodovací stromy jsou jedním z nejjednodušších nástrojů používaných pro podporu rozhodování. Jsou tvořeny sadou hierarchicky uspořádaných rozhodovacích pravidel, a jak již název napovídá, skládají se z kořene, který představuje celý soubor neroztřídněných dat, a postupně probíhá větvení do dalších uzlů, které ještě stále mohou obsahovat objekty více tříd, a je tedy nutné je dále dělit. Pokud nejsou již požadovány žádné podskupiny k dělení, v konkrétním uzlu strom přestane růst, o takovémto uzlu pak hovoříme jako o terminálním uzlu nebo také listu. Důležitým faktorem je správné nastavení rozsahu větvení pro každou proměnnou, v opačném případě totiž může dojít k situaci, kdy bude větvení příliš podrobné a nebude tedy možné vytvořit pro další třídy taková rozhodovací pravidla, aby nedocházelo k nehomogennímu dělení. Pokud je větvení stromu příliš podrobné, hovoříme o tzv. přeučení a jediným východiskem z této situace je tzv. prořezání stromu. Graficky lze rozhodovací strom znázornit pomocí orientovaného grafu. Dle topologie můžeme stromy dále dělit na binární a nebinární. Podle vstupní a výstupní proměnné lze dělit na kvalitativní a kvantitativní a tedy na klasifikační (kvalitativní proměnné) a regresní (kvantitativní proměnné) typ stromu.

3.1 Kriteriaální statistika

Před každým dělením musí být určeno, kterým kritériem se budou následující větve řídit, a tedy musí být určen takový atribut, který je schopen rozdělit proměnné v daném uzlu takovým způsobem, aby v každém poduzlu byly informace co nejhomogennější a mezi uzly co nejrozdílnější. K tomuto účelu se využívá kriteriaální statistiky. Existuje mnoho měření kriteriaální statistiky, která se navíc liší podle toho, jde-li o klasifikační či regresní strom. Pro regresní stromy používáme nejčastěji kritérium minima kvadratické chyby (*Least Square Deviation*) $Q(T)$:

$$\bar{y}_t = \frac{1}{N_t} \sum y_{i(t)}, \quad (3.1)$$

$$Q_t(T) = \frac{1}{N_t} \sum_{i=1}^{N_t} (y_i - \bar{y}_t). \quad (3.2)$$

Mezi kritéria, používaná pro klasifikační stromy, patří například Gini index, Entropie, klasifikační chyba, informační zisk atp.

$$\text{Gini index: } GI = 1 - \sum_{c=1}^J p_{tc}^2, \quad (3.3)$$

$$\text{Entropie: } H = - \sum_{c=1}^J p_{tc} \log_2 p_{tc}, \quad (3.4)$$

$$\text{Klasifikační chyba: } ME = 1 - \sum_{c=1}^J p_{tc}^2, \quad (3.5)$$

kde p_{tc} je podíl pozorování y_i s kategorií c v uzlu t z celkového počtu všech pozorování y_i v tomto uzlu neboli pravděpodobnost kategorie c v uzlu t .

Gini index (GI) je nejčastěji používaná kritériální statistika pro klasifikační stromy typu CART. Hodnota Giny indexu se rovná nule, pokud je v konečném uzlu pouze jediná kategorie proměnné Y a dosahuje maxima, pokud je v konečném uzlu v každé kategorii proměnné Y stejný počet pozorování. Ve chvíli, kdy dojde k rozdělení uzlu na dva dceřiné uzly, je GI spočítán pro každý dceřiný uzel. Celková hodnota Gini indexu pro dané rozdělení GI_{celk} je rovna váženému součtu všech GI indexů jednotlivých dceřiných uzlů. Vážení probíhá podle velikosti dceřiného uzlu. GI_{celk} tedy jednoduše spočítáme jako součet GI (i) dceřiných uzlů, které jsou vynásobeny příslušným podílem pozorování v daném dceřiném uzlu z celkového počtu pozorování v původním mateřském uzlu.

$$GI_{celk} = \sum_{i=1}^K \frac{N_i}{N_t} GI(i), \quad (3.6)$$

kde K je počet dceřiných uzlů (v případě binárního stromu se $K = 2$), N_t je počet pozorování v mateřském uzlu t a N_i jsou počty v dceřiných uzlech.

Entropie (H) dává velmi podobné výsledky jako Gini index. Dosahuje maxima, pokud jsou jednotlivé kategorie proměnné Y rovnoměrně zastoupeny v uzlech a minima pokud pozorování v uzlu náležejí pouze do jediné kategorie. Entropie je spočítána pro každý dceřiný uzel. Podobně jako u GI můžeme vyjádřit celkovou entropii H_{celk} pro dané dělení jako vážený součet entropií H (i) v dceřiných uzlech. Entropie je často používána v algoritmu C4.5.

$$H_{celk} = \sum_{i=1}^K \frac{N_i}{N_t} H(i). \quad (3.7)$$

Další kritérium, které lze použít pro rozdělení stromu je GAIN (information gain - informační zisk), který měří pokles v entropii.

$$GAIN_{celk} = H - \left(\sum_{i=1}^K \frac{N_i}{N_t} H(i) \right). \quad (3.8)$$

Poslední z výše uvedených kritériálních statistik je klasifikační chyba (ME). ME je podíl chybně klasifikovaných pozorování čili $1 - ME$ je celková přesnost stromu (podíl správně klasifikovaných pozorování). Klasifikační chyba je obvykle používána právě k finálnímu měření přesnosti klasifikačního stromu, proto je logické její použití jako kritériální statistiky. Celková klasifikační chyba pro dané dělení je opět váženým součtem ME v dceřiných uzlech.

$$ME_{celk} = H - \sum_{i=1}^K \frac{N_i}{N_t} ME(i). \quad (3.9)$$

3.2 Algoritmy růstu stromu

Binární stromy mají na výstupu uzlu vždy jen dvě větve, z tohoto důvodu mívají větší hloubku stromu než obecné stromy. Na druhou stranu jsou většinou přesnější a rychlejší. Obecné stromy mohou mít tedy libovolný počet hran na výstupu uzlu, oproti binárním bývají často menší a mají logičtější strukturu. K vytvoření rozhodovacích stromů se používají různé algoritmy, my zde popíšeme algoritmy binárního růstu CART a ID3. Pro obecné stromy potom popíšeme zejména algoritmus CHAID. Další možností je vytvořit strom tzv. učení s učitelem. Jde o proces tvorby rozhodovacího stromu, podle předem zvolených pravidel. Nevyužívá se tedy žádného algoritmu. V této práci je zvolen právě proces tvorby stromu pomocí učitele, jelikož se jedná o dosti řídký strom a použití algoritmu růstu by zde bylo nadbytečné.

3.2.1 Učení s učitelem

Je jednou z mnoha metod strojového učení pro učení funkce z množiny trénovacích dat. Tato data se skládají vždy ze vstupních a známých výstupních dat, jedná se tedy o učení pomocí indukce, kdy se ze známého výsledku snažíme odvodit operace, ke kterým došlo mezi vstupem a výstupem. Konkrétní příklad intuitivně vytvořeného stromu použitého v této práci a proces jeho učení, respektive nastavení klasifikačních úrovní, bude popsán v kapitole 7.1.

3.2.2 Algoritmus CART

Stromy tohoto typu jsou vhodné jak pro regresní, tak pro klasifikační úlohy a rostou na základě rekurzivního binárního dělení. Stejně jako u všech algoritmů vychází data z jednoho mateřského uzlu a následně se dělí, jelikož se jedná o binární dělení, do dvou dceřiných uzlů na základě určitého dělicího kritéria. Vzniklé uzly mohou být terminálními, za předpokladu, že data v těchto uzlech jsou konečná homogenní nebo se opět binárně dělí dle dalšího kritéria.

3.2.3 Algoritmus ID3

Tento algoritmus se používá pro nominální vstupy a výstupy. Na začátku větvení vycházíme opět z mateřského uzlu a na základě vhodných dělicích atributů jej dále rozdělujeme do dceřiných uzlů, na rozdíl od ostatních algoritmů zde může být konkrétní atribut dělení použit pouze jednou. K větvení se používá atributu s nejmenší průměrnou neuspořádaností. Konce větvení stromu dosáhneme, pokud již není volný žádný atribut, pokud strom dosáhl maximálního počtu uzlů, pokud je dosažena požadovaná přesnost, nebo pokud je nedostatek trénovacích dat. Chybí-li hodnota některého atributu, je nahrazena nejčastěji se vyskytující hodnotou v daném uzlu nebo nejčastěji se vyskytující hodnotou ve třídě, do které má být objekt klasifikován. Dojde-li k přeučení stromu, provádí se zpětné prořezávání stromu např. metodou *Reduced-Error Pruning*.

Volba atributu pro větvení se provádí na základě tzv. informačního zisku I . Nechť S je sada skládající se z datových záznamů o celkovém počtu s . Předpokládejme, že atribut s označením tříd má m rozdílných hodnot a definuje tak m rozdílných tříd C_i (pro $i=1, \dots, m$). Nechť s_i je počet záznamů patřících do třídy C_i . Očekávaná informační potřeba ke klasifikaci daného vzorku je dána vztahem,

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m p_i \log_2(p_i), \quad (3.10)$$

kde p_i je pravděpodobnost, že libovolný záznam patří do třídy C_i a je vypočítána jako poměr záznamů ve třídě s_i k celkovému počtu záznamů s .

Novější verzí vycházející z ID3 je algoritmus C4.5, který již disponuje metodou diskretizace pro numerické atributy, umožňuje práci s chybějícími hodnotami, převod na pravidla i prořezání. Tento algoritmus se stal standardem pro tvorbu rozhodovacích stromů. Další verzí je potom algoritmus C5.0, který je uživatelsky přívětivější.

3.2.4 Algoritmus CHAID

Stromy tohoto typu, jež jsou navrženy primárně pro kategoriální proměnné, bývají často využívány k různým marketingovým účelům. Jsou nebinárního typu, a tedy obsahují dělení do více než dvou dceřiných uzlů. Po prvním dělení také nemusí zbývat dostatek pozorování na vytvoření dalších „pater“ stromu. Tato technika je proto vhodnější pro větší datové soubory.

Kriteriální statistikou pro větvení je χ^2 -test, což vyplývá i z názvu metody. χ^2 -test je použit pro zjištění nezávislosti v kontingenční tabulce, která je tvořena kombinací kategorií závisle proměnné a prediktoru. Jsou-li Y a X nezávislé, má testová statistika přibližně Pearsonovo χ^2 rozdělení s $u = (r - 1)(s - 1)$ stupni volnosti, kde r je počet řádků a s je počet sloupců v kontingenční tabulce. Nezávislost v kontingenční tabulce znamená, že se obě proměnné navzájem neovlivňují v hodnotách, kterých nabývají. Hypotéza nezávislosti je zde nulovou hypotézou H_0 . Pearsonův χ^2 -test je často označován jako test dobré shody.

$$\chi^2 \text{-test: } \chi^2 = \sum_{i=1}^r \sum_{j=1}^s \frac{(p_{ij} - o_{ij})^2}{o_{ij}} \quad (3.11)$$

$$\text{Očekávané četnosti: } o_{ij} = \frac{R_i S_j}{n}, \quad (3.12)$$

kde i a j je označení řádků (resp. sloupců) v kontingenční tabulce, p_{ij} je pozorovaná frekvence, o_{ij} očekávaná frekvence, n je celkový počet pozorování, R_i je počet pozorování v řádku i , S_j je počet pozorování ve sloupci j .

3.3 Prořezání stromu

Složitost stromu určuje jeho velikost, pokud je strom příliš velký, je zbytečně složitý a ztrácí svou obecnou platnost. Pokud je strom příliš malý, většinou nezvládá postihnout všechna kritéria rozdělení dat a nedokáže tak data přesně zařadit, výsledkem je nesprávné či spíše nehomogenní rozdělení všech dat, tedy strom je nedokončen. Je tedy nutné najít optimální velikost stromu. Preferovaný přístup je nechat narůst velký strom, jako pravidlo se v tomto případě používá počet pozorování v uzlu (např. $n \leq 5$) a následně se strom prořeže. K určení optimální velikosti

stromu lze použít kritérium složitosti stromu. Mějme strom T_0 . Prořezáním určitého počtu koncových uzlů dostaneme strom T_1 . Kritérium složitosti stromu je rovno:

$$C_\alpha(T_1) = DT_1 + \alpha|T_1|, \quad (3.13)$$

kde $|T_1|$ je počet terminálních uzlů stromu a DT_1 je chyba stromu T_1 . Parametr $\alpha \geq 0$ vyjadřuje kompromis mezi velikostí stromu a jeho přesností. Pro každé α hledáme takový strom $T_\alpha \subseteq T_0$, který minimalizuje $C_\alpha(T)$. K určení odhadu α se používá křížová validace (krosvalidace).

4 ZPRACOVÁNÍ OBRAZU

Čerpáno z [10], [11]. Cílem předzpracování obrazu je zlepšit obraz z hlediska dalšího zpracování. Pod pojmem předzpracování obrazu si lze představit takové operace, které vedou ke zlepšení kvality obrazu a tedy k redukci nebo odstranění šumu a zkreslení v obraze a různých dalších nedokonalostí způsobených špatnými podmínkami při vytváření obrazu (jako je například nepříznivé osvětlení nebo špatná ostrost obrazu, způsobená pohybem scény nebo pohybem snímáče, a podobně, a také procesy, které potlačují či zvýrazňují určité rysy obrazu). Mezi tyto metody řadíme zejména bodové jasové transformace, geometrické transformace, lokální předzpracování, filtraci obrazu v kmitočtové oblasti, restauraci obrazu a matematickou morfologii 4.5. V této práci popíšeme pouze některé procesy předzpracování, a sice ty, které budou následně používány pro předzpracování v navrženém programu.

4.1 Jasová transformace

Jde o vratnou úpravu jasové stupnice obrazu, která si klade za cíl zvýraznit určité rysy obrazu. Bohužel některé jasové transformace způsobují také zvýraznění šumu. Tyto operace se dělí na bodové, lokální a globální, podle toho, čím se daná operace řídí. Bodové jasové transformace se řídí jasnem konkrétního jednoho pixelu, který pak přemění dle stanoveného vzorce na výsledný jas daného pixelu. Lokální transformace používá pouze malé, z pravidla čtyř až osmi-okolí daného bodu a dle těchto hodnot mění jas pouze v daném pixelu. Globální jasová transformace je taková transformace, která využívá hodnot jasu celého obrazu a aplikuje je na daný pixel v obraze. Nejčastějším využitím disponuje právě transformace bodová a to ve formě převodních funkcí nebo tzv. LUT tabulek. Sestavením převodních tabulek LUT udávajících závislost jasových hodnot výstupního obrazu na jasových hodnotách obrazu vstupního lze provést například gama korekci, redukci barev, prahování, úpravu jasu nebo kontrastu a podobně.

4.1.1 Lineární zvýraznění kontrastu scény

Jde o jasovou operaci, která si klade za cíl využít celou škálu jasové stupnice. Obraz je reprezentován určitými hodnotami jasu pro každý pixel, bohužel tyto hodnoty často nevyužívají celou škálu jasu, například z důvodu přexponování nebo podexponování obrazu či špatného nasvícení scény (např. v noci nebo při focení proti slunci). Těmto problémům se bohužel nemůžeme vyhnout. Program musí být schopen značku zachytit i při jízdě automobilu proti slunci nebo za šera. Rozsah jasu vstupního obrazu je tedy normován na celý rozsah možných jasových hodnot

0–255. Jedná se o početně nenáročnou matematickou operaci doloženou rovnicí 4.1 převzatou z [12].

$$O_{(x)} = \frac{255[I_{(x)} - I_{min}]}{I_{max} - I_{min}}, \quad (4.1)$$

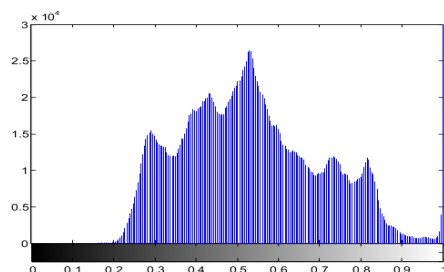
kde $O_{(x)}$ je výsledná hodnota jasu pro konkrétní pixel x . $I_{(x)}$ je počáteční hodnota jasu, kterou daný pixel na začátku disponoval. I_{min} je minimální hodnota jasu, která náležela danému snímku a I_{max} je maximální hodnota jasu, která náležela danému snímku.

V programu je lineární zvýraznění kontrastu použito pro snímky pořízené za špatných světelných podmínek, u kterých je značné riziko falešné nebo dokonce nulové detekce objektu. Příklad zvýraznění kontrastu je uveden na obrázku 4.1 spolu s příkladem ekvalizace histogramu, která provádí i vyrovnání histogramu. Pomocí

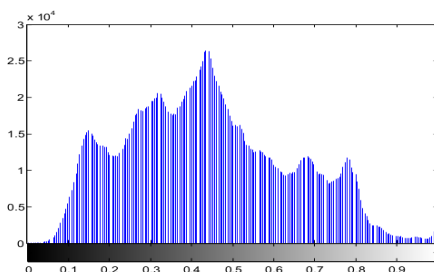


(a) Černobílý originál.

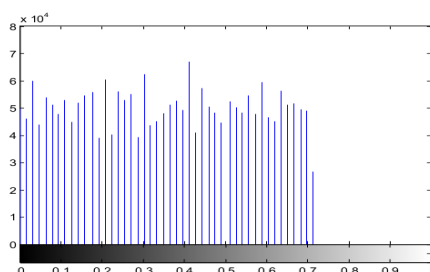
(b) Obráz po ekvalizaci histogramu.



(c) Histogram originálu.



(d) Histogram po roztažení.



(e) Histogram po ekvalizaci.

Obr. 4.1: Příklad roztažení a ekvalizace histogramu.

funkce *histeq* je možné provést již zmíněnou ekvalizaci 4.1.2, nicméně tento proces je pro naše účely poněkud nadbytečný. Přesto je v programu možné, zejména z důvodu zkvalitnění snímku pro uživatele, zvolit tuto variantu předzpracování obrazu.

Na obrázku 4.1a je vyobrazen šedotónový obraz vybrané scény a obrázek 4.1c ukazuje jeho histogram. Tato scéna byla vybrána záměrně, snímek je focen proti slunci, a proto je zde patrné výrazné přexponování obrazu. Je vidět, že histogram zdaleka nevystihuje všechny možné jasové úrovně, hlavně tmavé odstíny jasu jsou zde zastoupeny velmi řídky. Obrázek 4.1b pak dokládá výstupní černobílý obraz ekvalizace histogramu, jehož histogram je znázorněn na obrázku 4.1e. Zde je patrné roztažení jasové stupnice a vyrovnaní jasu. Obrázek 4.1d ukazuje histogram pouhého roztažení jasové stupnice. Černobílý obraz po roztažení jasové stupnice nebyl záměrně uveden, jelikož v něm nejsou patrné výrazné změny oproti originálu, na rozdíl od výstupu po ekvalizaci, kde je však spolu s výraznými změnami jasu patrné také zvýraznění šumu nebo spíše pro naše účely nepodstatných objektů v pozadí. Pro tento příklad bylo využito právě HSV modelu, ve kterém je, jak již bylo uvedeno v 2.1.3, velice snadné pracovat pouze s jasnem daného snímku.

4.1.2 Ekvalizace histogramu

Čerpáno z [6] a [13]. Vyrovnaní histogramu je technika používaná na šedotónový obraz, respektive upravující jasovou složku obrazu, která provádí, stejně jako předchozí, zvýraznění kontrastu. A tedy se snaží o co největší využití celého prostoru intenzit. Dalším cílem je dostatečně srovnat četnost jednotlivých úrovní intenzity a rozložit ji co možná nejrovnoměrněji. To vede často ke zvýraznění předtím špatně rozpoznatelných detailů obrazu. Histogram po aplikaci této metody lze vidět na obrázku 4.1e, kde je patrné, že některé jasové úrovně díky této technice již nejsou v obraze zastoupeny, to lze eliminovat aplikací dolní propusti. Samotný proces vyrovnaní intenzit v histogramu je popsán následující rovnicí:

$$I' = \frac{I}{XY} \sum_{i=I_0}^{i=I} H(i), \quad (4.2)$$

kde $H(i)$ je i -tá složka vektoru histogramu, X a Y jsou rozměry obrazu, I_0 je nejnižší intenzita původního obrazu a I je původní intenzita obrazového bodu.

4.2 Geometrická transformace

V kapitole je čerpáno zejména z [7]. Geometrická transformace objektu, ať již dvourozměrného či trojrozměrného, je jednou z nejčastějších a nejdůležitějších operací užívaných v počítačovém vidění. Mezi tyto operace řadíme posunutí (translaci), rotaci, změnu měřítka (škálování), zkosení a různé kombinace těchto operací nazývané složené transformace. Samotná transformace je prováděna v prostoru *homogenních souřadnic*. Převod z kartézských do homogenních souřadnic je popsán vzorcem 4.3. V této kapitole se budeme věnovat geometrickým transformacím, které jsou aplikovány na 2D objekty, zejména pak změně měřítka, rotaci, zkosení a jejich složeným transformacím, více viz [6] nebo [7].

$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \mathbf{A}_{3 \times 3} P = \mathbf{A} \begin{bmatrix} x \\ y \\ w \end{bmatrix}, \quad (4.3)$$

4.2.1 Změna měřítka

Změna měřítka ovlivňuje velikost a polohu transformovaného objektu ve vztahu k počátku souřadnic. Touto operací lze provést zmenšení či zvětšení objektu, podle toho jaké hodnoty nabývají koeficienty měřítka. Jsou-li tyto koeficienty v intervalu $(0, 1)$, dochází ke zmenšení objektu a zároveň k jeho posuvu směrem k počátku souřadnic. Náleží-li koeficienty některé z hodnot v intervalu $(1, \infty)$, dojde k zvětšení a zároveň posunutí směrem od počátku souřadnic. Bude-li znaménko koeficientu záporné, dojde ke stejným procesům, avšak ve směru opačném. Následující matice popisuje transformaci 2D objektu.

$$\mathbf{S}(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.4)$$

kde s_x je koeficient změny měřítka ve směru osy x a s_y je koeficient změny měřítka ve směru osy y .

Následující příklad znázorňuje operaci změny měřítka a zároveň také demonstruje smysl použití této operace za účelem identifikace daného objektu.



(a) Detekovaná značka.



(b) Vzor pro určení velikosti.



(c) Výsledek změny měřítka značky.

Obr. 4.2: Příklad změny měřítka objektu podle velikosti vzoru.

Na obrázku 4.2 je demonstrován příklad změny měřítka objektu. Snímek 4.2a ukazuje nalezenou značku v originálních rozměrech, značka je vyfocena jako pohled zprava. Další snímek 4.2b představuje vzor pro konkrétní tvar značky vybraný náhodně z databáze výstražných značek, který zde slouží pouze informativně k nastavení správné svislé a vodorovné délky detekované značky. Poslední obrázek 4.2c vyobrazuje výslednou změnu měřítka. Značka je však stále nakloněná a změnou měřítka do jisté míry také deformována. Z uvedeného příkladu je tedy patrné, že pro lepší výsledek, bude potřeba aplikovat na objekt další geometrické transformace (zde například rotaci a zkosení).

4.2.2 Rotace

Rotací objektu je zde myšleno posunutí objektu a všech jeho bodů o určitý úhel, který svírá s počátkem souřadnic (levý dolní roh). Úloha rotace je velmi prostá, ať je objekt jakkoli natočen, vycházíme z předpokladu, že svírá s počátkem nulový úhel. Pro rotaci objektu tedy jednoduše zvolíme úhel α . Dle znaménka úhlu určíme směr rotace po směru (kladné) nebo proti směru (záporné) hodinových ručiček. A následně dosadíme do transformační matice 4.5.

$$\mathbf{R}(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.5)$$

inverzí dané matice $\mathbf{R}(\alpha)$ získáme výslednou rotaci shodnou s rotací o stejný ovšem záporný úhel α . Příklad rotace je uveden na obrázku 4.3 spolu s procesem zkosení a změnou měřítka.

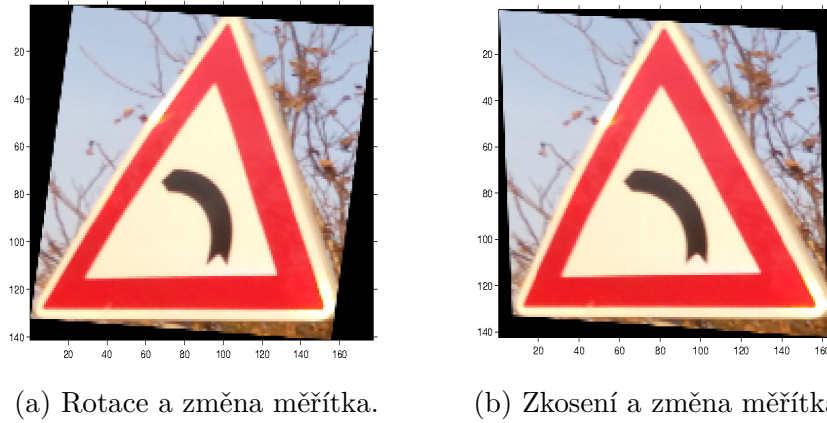
4.2.3 Zkosení

Zkosení provádíme podle osy x, respektive y. Podle orientace tedy dosazujeme za koeficient zkosení do jedné z následujících matic 4.6. Inverzí matic získáme zkosení v záporném směru dané osy.

$$\mathbf{Sh}_x(sh_x) = \begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Sh}_y(sh_y) = \begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.6)$$

4.2.4 Složené transformace

Složením různých transformací získáme jednu matici popisující všechny použité operace. Jedná se o maticové násobení, proto záleží na pořadí daných operací. Bude tedy rozdíl, pokud provedeme posunutí a následně operaci rotace kolem počátku, nebo tento proces provedeme v opačném pořadí (resp. pokud bychom chtěli stejný výsledek, museli bychom vypočítat nové koeficienty v závislosti na první operaci). Pro naše účely je vhodné sestavit matici složenou ze zkosení a změny měřítka, ne nutně v tomto pořadí. Tuto operaci popisuje následující rovnice 4.7.



Obr. 4.3: Příklad geometrických transformací.

$$\mathbf{V}(sh_x, sh_y, s_x, s_y) = \mathbf{Sh}_x(sh_x) \mathbf{Sh}_y(sh_y) \mathbf{S}(s_x, s_y) =$$

$$\begin{bmatrix} s_x + s_x sh_y sh_x & s_y sh_x & 0 \\ s_x sh_y & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.7)$$

kde s_x a s_y jsou koeficienty změny měřítka, sh_x a sh_y jsou koeficienty zkosení, ve směru dané osy. Takto složenou transformaci demonstruje obrázek 4.3b.

Na obrázku 4.3 jsou znázorněny složené geometrické transformace objektu z 4.2a. Hlavně transformace zkosení a změna měřítka dokládá výrazné zlepšení geometrických vlastností objektu. Následné porovnání se vzorem tak bude již velmi snadné. Uvedené transformace byly provedeny prostřednictvím programu Matlab s využitím funkce *affine2d*. Nejprve byla detekovaná značka převedena do binárního obrazu, zde byly změřeny délky stran a určeny souřadnice krajních bodů. Z poměrů velikostí stran nalezené značky a vzoru byly určeny koeficienty změny měřítka s_x a s_y . Následně byl určen úhel α , který definuje vodorovný sklon značky a úhel β , který definuje úhel rotace k ose y. Zkosení bylo vypočteno jako rozdíl těchto dvou úhlů, tedy $q = \alpha - \beta$. Takto normalizovaná značka je již připravena k aplikaci tzv. template matchingu.

4.3 Detekce hran

Čerpáno z [14] a [15]. Hrana jako taková indikuje hranice objektu v obraze. Detekce hran je základní operací pro detekci významných lokálních změn intenzity v obraze, často se také používá k ostření (zvýraznění extrémů) obrazu. Každá tato změna je měřena gradientem obrazu, a protože je obrazová funkce $f(x, y)$ dvourozměrná, je její gradient vektor:

$$\nabla f(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix}, \quad (4.8)$$

kde složka G_x je parciální derivací obrazové funkce $f(x, y)$ podle osy x a G_y je parciální derivací obrazové funkce dle osy y . Jedná se tedy o hrany průchodu první derivací. Protože složky G_x a G_y udávají změnu obrazové funkce pouze podél osy x resp. y , je třeba vypočítat velikost G a směr θ gradientu.

$$\begin{aligned} G[f(x, y)] &= \max \{|G_x|, |G_y|\}, \\ \theta(x, y) &= \arg(G_x, G_y). \end{aligned} \quad (4.9)$$

4.3.1 Hranové detektory aproximující první derivaci obrazové funkce

Těchto detektorů existuje celá řada. My zde popíšeme pouze Sobelův, Prewittův a Cannyho hranový detektor. Tyto operátory nejsou invariantní vůči rotaci, proto je nutné počítat jednotlivé odezvy s různě otočenými maskami.

Prewittův hranový detektor

Společně se Sobelovým operátorem je považován za jeden ze základních operátorů pro detekci hran. Gradient je odhadován pro čtyř-okolí bodu, tedy maska je maticí 3×3 . Tento detektor používá celkem osm masek. Z těchto masek se vybírá pouze ta, které odpovídá největší modul gradientu. Základní maska je popsána následujícím vyjádřením 4.10, ostatní masky jsou pouze rotací této, vždy o 45° .

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}. \quad (4.10)$$

Sobelův hranový detektor

Sobelův operátor disponuje stejným mechanismem jako Prewittové, jediný rozdíl je ve váze koeficientů u sousedů ve čtyř-okolí. Je opět definován maskou 4.11 pro tzv. jižní hrany.

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (4.11)$$

Cannyho hranový detektor

Je jedním z nejpoužívanějších hranových detektorů, vykazuje dobré výsledky i se značně zašuměným obrazem. Mezi podmínky optimálního chodu tohoto detektoru patří:

1. minimální chybovost – místa, která nejsou hranami, se nesmí detekovat,
2. přesnost – poloha hrany musí být detekována s maximální přesností,
3. jednoznačnost – hrany se detekují pouze jednou.

Postup pro detekci má čtyři kroky. Nejprve se eliminuje šum pomocí Gausovského filtru, tento filtr je nejčastěji realizován konvoluční maskou. Dále se určí gradienty z první derivace, jejich velikost a směr. Toto se provede aplikací Sobelova operátoru. Následně se vyberou lokální maxima zjištěných gradientů. V tomto kroku je označen pouze ten bod, který má větší hodnotu gradientu, než jeho sousední body, nacházející se v okolí kolmém na tento bod, tzv. *thinning*. Posledním krokem je tzv. *thresholding* (neboli prahování s hysterezí). Úkolem je zde ohodnocení významu nalezených hran, aby nedošlo k mnohonásobné odezvě hrany. Vzhledem k přítomnosti hran získaných z šumu je zapotřebí určit dva prahy T_1 a T_2 , jejichž úkolem je rozdělit množinu získaných hran tak, aby byly potlačeny hrany získané z šumu, ale ostatní hrany byly zachovány, více viz [16].

4.4 Segmentace

Čerpáno z [17]. Cílem segmentace je oddělení objektů zájmu od pozadí scény, tím redukovat data obsažená v obraze a usnadnit tak další analýzu obsahu obrazu. Dalším krokem po segmentaci je popis segmentovaných dat, která dále můžeme kategorizovat. Segmentace často pracuje s globálními znalostmi obrazu, jako je například barva, jas, velikost a další vlastnosti objektů. Výsledkem segmentace by měl být soubor vzájemně se nepřekrývajících oblastí, které v ideálním případě jednoznačně korespondují s objekty. Problémy segmentace může způsobit mnoho faktorů,

jako je například velká složitost scény, špatné světelné podmínky, šum nebo překrývající se objekty. Mezi metody segmentace řadíme: segmentaci prahováním (která je zde blíže specifikována, zejména kvůli jejímu následnému využití v programu), segmentace založené na detekci hran a segmentace založené na hledání oblastí či Houghova transformace 5.3.

4.4.1 Segmentace prahováním

Objekty či oblasti jsou charakterizovány jasem nebo barvou. Vychází se z předpokladu, že objekty mají rozdílné vlastnosti oproti pozadí. Prahování je transformace vstupního obrazu na výstupní často binární obraz, kde obrazovým elementům náležícím objektu (mají větší hodnotu nežli práh) je přiřazena hodnota 1, ostatní elementy (elementy pozadí) mají hodnotu 0. Prahování takto popsané se nazývá prosté. Dalšími možnostmi jsou prahování s více prahy nebo částečné či adaptivní (lokální) prahování.

4.5 Matematická morfologie

Teorie převzata z [18] a [19]. Matematická morfologie (dále jen MM) byla původně určena k analýze pouze binárních obrazů, později však byla zobecněna i pro analýzu šedotónových obrazů. Využívá se zejména k detekci hran, redukci šumu nebo například k segmentaci. V následných kapitolách bude pozornost věnována zejména binární MM.

4.5.1 Operátory MM

Základními operátory morfologických operací jsou:

- dilatace,
- eroze,
- otevření,
- uzavření,
- skelet,
- tref či miň (hit or miss),
- ztenčování,
- a zesilování.

V této kapitole však popíšeme pouze operace eroze, dilatace, otevření a uzavření (popis všech morfologických operací je k dispozici v [18]). Morfologická transformace je operátorová relace zpravidla dvou bodových množin, z nichž první množinou např.

X je samotný binární / šedotónový obraz. Druhá množina např. B je poté nazývána sondou (jedná se o předem definovaný strukturální element).

Dilatace

Jedná se o tzv. *Minkowského součet* a jde o sjednocení posunutých bodových množin:

$$X \oplus B = \bigcup_{b \in B} X_b, \quad (4.12)$$

lze ji také vyjádřit funkcí δ :

$$\delta_B(X) = X \oplus B, \text{ kde } \{p \in \mathbb{E}^2 : p = x + b, x \in X \wedge b \in B\}, \quad (4.13)$$

kde X je binární obraz, složený z bodů x , podrobený dilataci strukturálním elementem B , skládajícím se z bodů b a p je nový obrazový bod vzniklý dilatací.

Vlastností dilatace je komutativnost, asociativnost, rostoucí transformace a invariantnost vůči posunu.

Eroze

Minkowského rozdíl:

$$X \ominus B = \bigcap_{b \in B} X_b. \quad (4.14)$$

Pro každý bod obrazu p je ověřeno, zda pro všechna možná $x + b$ leží výsledek v obraze X . Pokud ano, je výsledek logická 1, v opačném případě 0.

$$X \ominus B, \text{ kde } \{p \in \mathbb{E}^2 : p = x + b \in X \forall b \in B\}. \quad (4.15)$$

Dilatace a eroze jsou vzájemně duální operace, tedy se nejedná o vzájemně inverzní operace. Vlastností eroze je antiextenzivita, invariantnost vůči posuvu, zachování inkluzí (laicky řečeno výsledkem operace je vždy původní obraz nebo jeho podmnožina) a další. Pomocí eroze lze jednoduše vytvořit obrys objektu:

$$\partial X = X \setminus (X \ominus B), \quad (4.16)$$

kde ∂X je obrys obrazu X , který vznikne doplnkem výsledného obrazu. Respektive jde o ty body, které jsou erozí podle B odstraněny.

Otevření

Jedná se o erozi následovanou dilatací:

$$X \circ B = (X \ominus B) \oplus B. \quad (4.17)$$

Morfologické otevření odstraňuje šum a odděluje objekty spojené tenkou hranicí. Používá se zejména pro snížení množství detailů v obraze. Jeho vlastností je antiextenzivnost a idempotentnost (opakovaná aplikace otevření již nemění výsledek).

Jedná se o tzv. morfologický filtr, jež definuje zachování uspořádání a právě zmíněná idempotentnost čili konvergence během jedné iterace.

Uzavření

Dilatace následovaná erozí:

$$X \bullet B = (X \oplus B) \ominus B. \quad (4.18)$$

Morfologické uzavření je duální operací k otevření. Jedná se taktéž o jeden z nejdůležitějších morfologických filtrů, který spojuje blízké objekty a zaplňuje díry. Jeho vlastností je extenzivnost a již zmíněná idempotentnost.

5 POPIS A IDENTIFIKACE OBJEKTŮ

Čerpáno výhradně z [20] a [21]. Je snahou o získání příznaků ze segmentovaných dat, kde základní myšlenkou je volba vhodných deskriptorů, které musí vystihovat charakteristické rysy popisovaných objektů. Vzhledem k procesu lidské rekognice objektů je metoda využití deskriptorů velice závislá na subjektivním pohled programátora. Lidský mozek totiž k odlišení různých tvarů od sebe neužívá klasifikaci založenou na explicitním výpočtu hodnot příznaků. Hodnoty musí být zvoleny tak, aby bylo dělení objektů co možná nejhomogennější a počet různých tříd co možná největší. K tomuto účelu slouží již zmíněné radiometrické deskriptory 5.2. Ty mohou být založený na regionech nebo na hranicích objektu. Pokud je potřeba globálního popisu objektů v obraze, je před samotnou aplikací popisu nutné tyto oblasti odlišit. K tomu slouží tzv. identifikace oblastí. Další možností je využití Houghovy transformace k detekci přímek či různých geometrických entit.

5.1 Identifikace oblastí

V této kapitole předpokládáme vstupním obrazem vysegmentovaný binární obraz zvolené scény. Snahou je označit každý objekt unikátním číslem. To může být provedeno dvěma základními algoritmy:

1. rekurzivní vyplňování,
2. dvouprůchodový algoritmus.

Protože je v programu s výhodou využito funkce *bwlabel*, bude popsán pouze algoritmus dvouprůchodový (více k identifikaci oblastí: [21]).

5.1.1 Dvouprůchodový algoritmus

V prvním kroku jsou označeny souvislé části oblastí pomocí posunu masky, ta může být maskou čtyř či osmi-okolí. To se děje následovně: nejprve je obraz procházen bod po bodu od počátku po řádcích až do posledního pixelu obrazu. Pokud je nalezen pixel s hodnotou log. 1, je označen unikátním číslem i . Za předpokladu, že je následující bod logickou nulou, další bod log. 1 v řádku bude označen indexem $i_{max} + 1$, v opačném případě je další jedničky přiřazen stejný index i . Na dalším řádku jsou maskou opět procházeny jednotlivé body. Pokud maska narazí na bod log.1, který je v sousedství s již označeným bodem (odpovídá masce), přidělí mu stejný index. V opačném případě je přidělena opět hodnota $i_{max} + 1$). Toto se děje dokud algoritmus nedojde k poslednímu bodu v obraze. Ve druhém kroku algoritmu

jsou vyhledány konfliktní oblasti (tj. oblasti, u kterých se podle masky nemohlo jednoznačně rozhodnout, jaký index přidělit) a je jim přiřazen index s vyšší prioritou. Nakonec jsou přeznačeny všechny pixely s nižší prioritou. Výsledkem je pak obraz, v němž hodnota i_{max} odpovídá počtu oblastí v obraze.

5.2 Radiometrické deskriptory

Deskriptory mohou být založeny na hranicích objektu nebo na regionech. Příznaky založené na hranicích objektu mohou být reprezentovány binárním obrazem (implicitně) nebo geometrickými entitami (explicitně), více viz [21]. Následující kapitola blíže specifikuje příznaky založené na regionech.

5.2.1 Deskriptory založené na regionech

Popisují metrické vlastnosti vypočtené z plošného rozložení pixelů konkrétního objektu, jako například velikost, obsah, kompaktnost a další, které jsou blíže specifikovány níže. V navrženém algoritmu jsou tyto příznaky použity a díky funkci *regionprops* je popis objektů velice usnadněn. Více v části Implementace, kapitola Popis objektů 7.4.

Deskriptory použité v sestaveném algoritmu

Velikost (*Area*) Jde v podstatě o obsah objektu, respektive o počet pixelů náležících objektu. Výsledkem je hodnota obsahu.

Obvod (*Perimeter*) Jedná se o počet hraničních pixelů. Deskriptor je závislý na rozlišení snímku. Výsledkem je hodnota obvodu.

Konvexnost (*Solidity*) V intervalu $<0,1>$ vyjadřuje míru podobnosti objektu ke své konvexní schránce. Pokud všechny body spojnice dvou libovolných bodů objektu náleží tomuto objektu, je objekt konvexní a je mu přidělena hodnota jedna.

Hlavní osa (*MajorAxisLength*) Délka hlavní osy elipsy, jejíž centrální moment druhého řádu (rozptyl) má stejnou hodnotu jako centrální moment druhého řádu popisovaného objektu. Výsledkem je hodnota délky hlavní osy elipsy.

Vedlejší osa (*MinorAxisLength*) Délka vedlejší osy elipsy, jejíž centrální moment druhého řádu má stejnou hodnotu jako centrální moment druhého řádu

popisovaného objektu. Výsledkem je hodnota délky vedlejší osy elipsy.

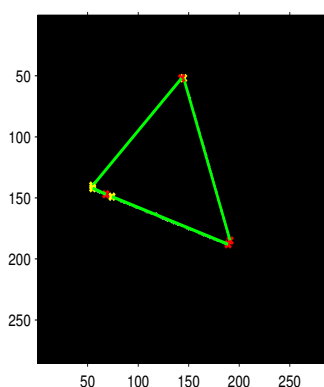
Výstřednost (*Eccentricity*) Výstřednost elipsy, jejíž centrální moment druhého řádu odpovídá centrálnímu momentu druhého řádu popisovaného objektu. Výsledkem je hodnota výstřednosti v rozsahu $<0,1>$, kde nula značí výstřednost kruhu a 1 výstřednost úsečky.

Kompaktnost Vyjadřuje míru podobnosti objektu ke kruhu. Vypočítá se jako $\text{obvod}^2 / \text{velikost}$.

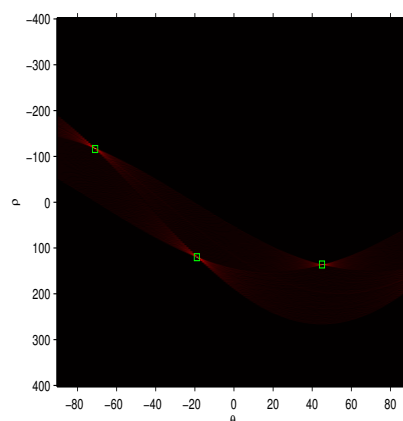
Hodnoty jednotlivých deskriptorů většinou svou přesností odpovídají rozlišení obrazu. Čím větší (robustnější) objekty popisujeme, tím přesnější výsledky by měly příznaky ukazovat. Pokud deskriptor vrací hodnotu velikosti, jedná se o velikost v pixelech. Výše popsané deskriptory nejsou zdaleka všechny, které je možné získat pomocí funkce *regionprops*. Informace o dalších příznacích jsou k dispozici zde [21].

5.3 Houghova transformace

Čerpáno z [6] a [22]. Jedná se o matematickou metodu používanou k hledání přímk v obraze, ale lze ji také využít k hledání různých, analyticky popsatelných tvarů (kružnice, elipsa, trojúhelník apod.). Nejčastěji se transformace aplikuje na vyhranovaný binární obraz často pomocí *Cannyho* detektoru. Velice důležité je, aby byly hrany co nejtenčí (nejlépe jeden pixel) a obraz byl co možná nejčistší, v opačném případě je, při nedostatečně přesném nastavení hlasování, na výstupu přítomno velké množství falešných přímk. Následující obrázek zobrazuje proces detekce přímek, resp. úseček, ve vyhranovaném binárním obraze trojúhelníka 5.1.



(a) Nalezené úsečky.



(b) Akumulátor.

Obr. 5.1: Příklad Houghovy transformace.

Na obrázku 5.1a je zobrazen trojúhelník vyhranovaný pomocí *Cannyho* hránového detektoru, zelené linie zde vyznačují nalezené úsečky, ohraničené vždy žlutým a červeným bodem (počáteční a koncový bod). Druhý obrázek 5.1b znázorňuje tzv. akumulátor, jedná se Houghův prostor, do kterého jsou mapovány jednotlivé body tvořící výsledné úsečky, ty jsou zde ohraničeny zeleným čtvercem. V tomto prostoru jsou body mapovány na křivky a úsečka (přímka) je zobrazena jako průsečík těchto křivek. Jak můžeme vidět na obrázku 5.1a, metoda transformace hran na přímky často vytvoří na jedné hraně více úseček, toto lze do jisté míry eliminovat vhodným nastavením parametrů transformačního algoritmu. V některých případech je však nutno nalezené oblasti v akumulátoru porovnat a určit, zda ještě patří k jedné hraně, nebo se jedná již o jinou hranu.

Rovnice přímky definovaná v kartézské soustavě souřadnic $y = ax + b$ je nejprve vyjádřena v polárních souřadnicích $\rho = x \cos(\theta) + y \sin(\theta)$. Poté je vytvořeno akumulátorové pole s počtem dimenzí rovným počtu neznámých v hledaném útvaru (zde dvě, např. v případě kružnice 3). Následně je pro každý bod (x, y) v prohledávaném obraze a pro všechny hodnoty úhlu θ , tj. úhel mezi normálou přímky a osou X , vypočtena délka této normály ρ . Hodnota je poté zapsána na konkrétní pozici (ρ, θ) v akumulátoru. Nejvyšší hodnota v akumulátoru pak koresponduje s nejdůležitějšími přímkami v obraze.

Houghova transformace se dnes v počítačovém vidění využívá velmi často. Příkladem mohou být různé algoritmy hledání určitého objektu v obraze například SPZ vozidla nebo detekce očí a podobně. Nutno říci, že pokud se snažíme hledat pouze jeden určitý tvar, jeví se tato možnost jako nejlepší řešení. V případě hledání více

různě popsanych geometrických obrazců je však tato metoda v některých případech spíše nepraktická a velice časově náročná.

Existují i další modernější principy aplikace Houghovy transformace jako takové, díky nim je možné hledat jakýkoli tvar bez potřeby znalosti jeho analytického popisu. Patří sem například tzv. *generalized Hough transform* (Zobecněná Houghova transformace). Prakticky je vytvořen popis hranic objektu v závislosti na zvoleném referenčním bodu, nejčastěji je tímto bodem těžiště. Tento je pak uložen do tzv. R-tabulky, dle které se pak celá transformace řídí. Nabízí se také možnost využít tzv. *randomized Hough transform*, která řeší problém s dlouho trvajícím hlasováním pro každý nenulový bod v obraze nebo např. *Probabilistic Hough Transform* (více viz [23]).

6 TECHNIKY ROZPOZNÁNÍ

V této kapitole popíšeme využití techniky rozpoznání, jimiž jsou tzv. *Template matching* a algoritmus *SURF*. Obě tyto metody jsou schopny do jisté míry úspěšně nalézt hledaný objekt v obraze, bez větší potřeby složitějšího předzpracování obrazu a jeho segmentace, nicméně je tento postup vhodný pouze za předpokladu, hledáme-li v obraze jeden konkrétní objekt. Metoda porovnání se vzorem je navíc velice závislá na geometrických vlastnostech objektu. Při větším počtu vzorů a nutnosti vzájemného porovnání příznaků by byl takto zvolený postup příliš časově náročný a dosti nepřesný, více v části Implementace, kapitola Rozpoznání.

6.1 Template matching

Čerpáno z [20]. Metoda porovnání se vzorem (maskou), která je založena na výsledku vzájemné korelace dvou signálů. Její použití je vhodné, zejména pokud s určitou pravděpodobností víme, že hledaný vzor se v obraze skutečně nachází a chceme pouze určit přesnou pozici tohoto objektu. Úspěšnost této metody je navíc dosti ovlivněna geometrickými vlastnostmi masky. Existují dva přístupy k template matchingu:

Features-based – porovnává shodnost hran a rohů.

Template-based – porovnává shodu celé masky a částí obrazu, například při korelaci jasových složek jsou rozhodujícím faktorem podobnosti jednotlivé hodnoty jasu konkrétních pixelů.

Vzory mohou být hledány v celém obraze takzvanou křížovou korelací *cross correlation* (protože se pohybujeme ve dvou a více rozměrných prostorech, používáme dvourozměrné korelace), v matlabu *xcorr2* nebo pomocí klasické korelace porovnávány přímo s podezřelými oblastmi 7.7. V matlabu *corr2*.

6.1.1 Korelace

Klasická dvourozměrná korelace je využita v navrženém algoritmu a její nespornou výhodou oproti křížové korelaci, popsané níže, je nižší časová náročnost. Je založena na výpočtu korelačního koeficientu R dvou porovnávaných signálů **shodné velikosti**:

$$R = \frac{\sum_m \sum_n (X_{mn} - \bar{X})(H_{mn} - \bar{H})}{\sqrt{(\sum_m \sum_n (X_{mn} - \bar{X})^2)(\sum_m \sum_n (H_{mn} - \bar{H})^2)}}, \quad (6.1)$$

kde \bar{X} je průměr jasových hodnot určený ze všech prvků matice \mathbf{X} a \bar{H} je průměr jasových hodnot určený ze všech prvků matice \mathbf{H} . X_{mn} a H_{mn} značí konkrétní hodnotu jasu v daném pixelu.

6.1.2 Křížová korelace

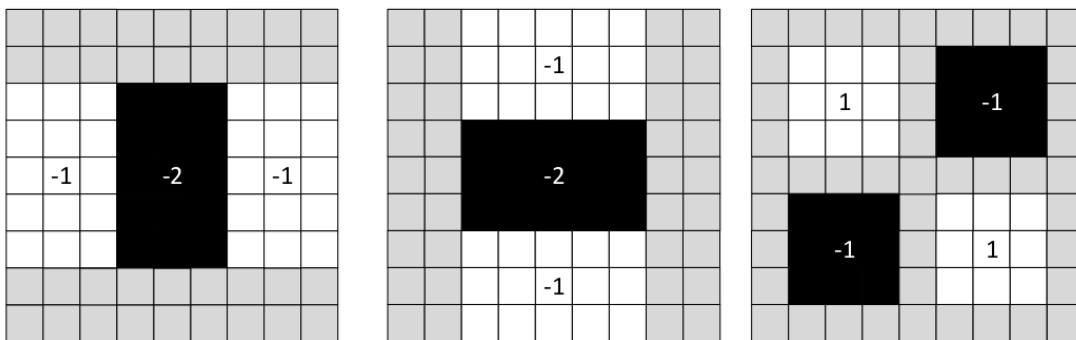
Pro jednoduchost předpokládejme vzor, reprezentovaný pouze jasovou složkou \mathbf{H} o rozměrech $P \times Q$, který je podmnožinou tzv. zdrojového obrazu, taktéž reprezentovaného jasovou maticí \mathbf{X} o rozměrech $M \times N$. Pro nalezení hledaného objektu v celém obraze použijeme již zmíněnou dvourozměrnou křížovou korelaci tohoto obrazu a předem definovaného vzoru $\mathbf{C} = \mathbf{X} * \mathbf{H}$. Výsledkem je matice podobnostních koeficientů \mathbf{C} o rozměrech $M + P - 1 \times N + Q - 1$.

6.2 Algoritmus SURF

Čerpáno z [24]. *Speeded-Up Robust Features* (SURF) jehož hlavní výhodou je rychlost. Algoritmus je inspirován metodou SIFT, která je blíže popsána ve [24]. K vysoké rychlosti nejvíce přispívá použití integrálních obrazů v konvolucích.

6.2.1 Detekce významných bodů

Významné body jsou detekovány jako maxima (mají největší hodnotu ve svém okolí) uvnitř scale-space vytvořeného konvolucí filtračních jader o různých velikostech s konkrétním obrazem, kde jsou body porovnány se svým 26okolím (8 sousedů z konkrétní vrstvy +18 sousedů z vrstev sousedních). Lokalizace takto nalezeného bodu je poté zpřesněna Taylorovým rozvojem [24]. Hessova matice je použita také k odstranění hran obrazu a jedná se o matici druhých parciálních derivací konvoluce vstupního obrazu s Gaussovou funkcí. Protože je metoda nezávislá na změně měřítka, je do výpočtu Hessovy matice zahrnuto také měřítko. Aby byla metoda co nejrychlejší, jsou k výpočtu matice použity konvoluce vstupního obrazu s filtry znázorněnými na obrázku 6.1, které aproximují druhou derivaci v diskrétní formě.



Obr. 6.1: Aproximace druhé derivace Gaussovy funkce (zleva doprava) podle x, y a dle xy. Obrázek je inspirován [24].

Determinant Hessianu je pak vypočten následovně

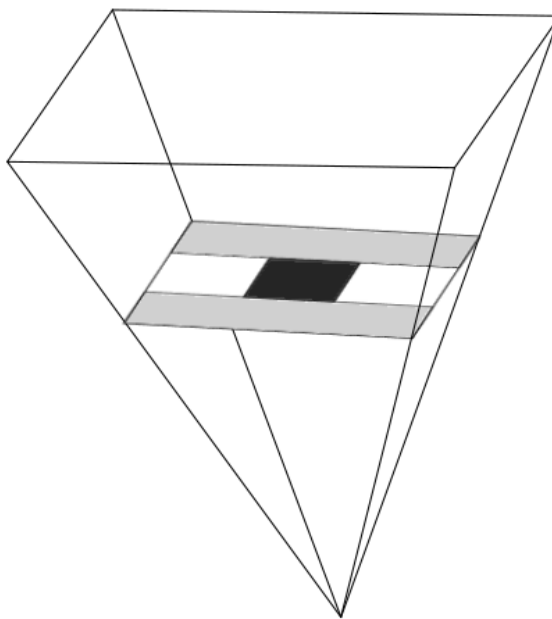
$$\det(\mathbf{H}_{approx.}) = D_{xx}D_{yy} - (wD_{xy})^2, \quad (6.2)$$

kde D_{xx} , D_{yy} a D_{xy} jsou odezvy obrazu na uvedené filtry. Ty jsou normalizovány vzhledem ke své velikosti, a proto je zavedená váha w konstantní pro všechna měřítka.

6.2.2 Scale-space

Aby byla metoda nezávislá na změně měřítka, je nutné vytvořit prostor možných měřítek z jednotlivých obrazů vzniklých filtrací. Konvoluce pro různě velká filtrační jádra je urychlena implementací integrálního obrazu. Obraz není třeba podvzorkovat, což by mělo za následek tzv. Aliasing. Scale-space je znázorněn na obrázku 6.2 a dělí se na osm oktáv, kde základem je odezva na filtrační jádro 9×9 znázorněné na obrázku 6.1. Tato jádra pak postupně narůstají na rozměry 15×15 , 21×21 a 27×27 . Poté je již splněna podmínka, kdy nejvyšší vrstva má více než dvojnásobné měřítko základní vrstvy. Filtrační jádra v základní oktávě narůstají o $2/3$ velikosti základního jádra. Druhá oktáva je pak tvořena druhou vrstvou oktávy předešlé, čili 15×15 , a krok je dvojnásobný. Oproti první oktávě tedy jádra narůstají o 12.

Po sestavení filtračních jader různých velikostí je vypočten determinant Hessovy matice, tedy konvoluce pro každé jádro a vstupní obraz zvlášť. Vyniklé odezvy (obrazy) pak představují finální měřítkově nezávislou reprezentaci vstupního obrazu. Dalším krokem je pak lokalizace významných bodů v těchto obrazech.



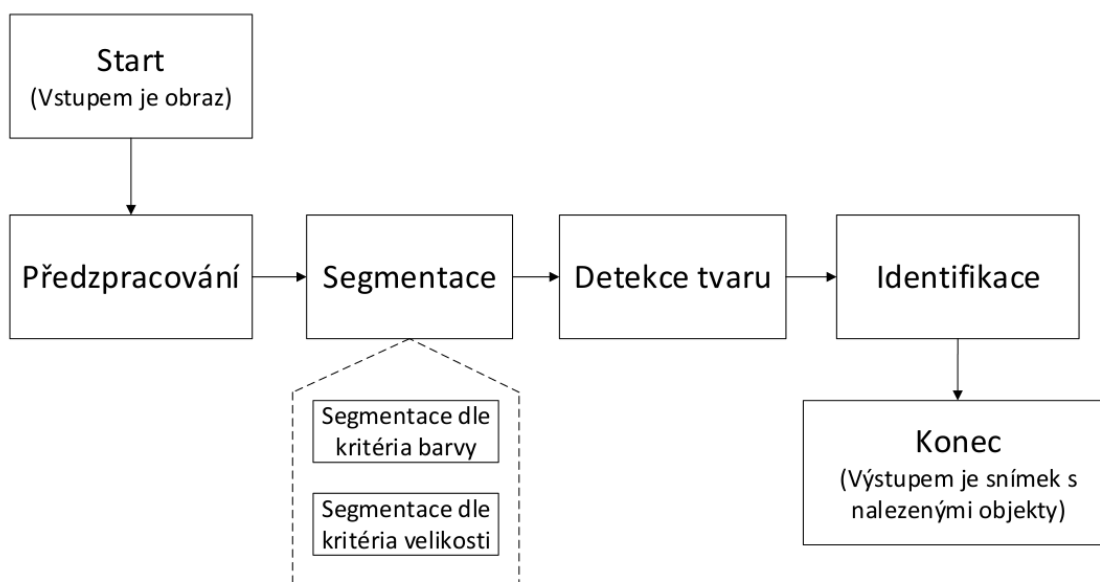
Obr. 6.2: Filtrační jádro metody SURF. Obrázek je inspirován [24].

6.2.3 Přiřazení orientace významným bodům

Přiřazení orientace významnému bodu má za následek nezávislost celé metody na rotaci. V případě metody *SURF* je na rozdíl od metody *SIFT* využito odezvy na Harrovu vlnku v kruhovém okolí 6 s (s je měřítko v konkrétním patře scale-space) významného bodu v konkrétním patře měřítkového prostoru. Ta je opět aproximována pomocí obdélníkových filtrů x a y . Takto vypočtené odezvy jsou o to rychlejší, díky opětovnému využití integrálního obrazu pro okolí významného bodu na Harrovu vlnku. Odezvy jsou pak označeny jako d_x a d_y a jsou váženy koeficienty Gaussova kruhového okna. Následně jsou koeficienty zakresleny do kruhového prostoru, který je navíc rozdělen na šestinové výseče (tedy $\pi/3$). V těchto výsečích jsou poté definovány vektory $\sum d_x$ a $\sum d_y$. Výseč s největším vektorem pak představuje dominantní orientaci významného bodu. Následně jsou vytvořeny deskriptory nalezených maxim s tím rozdílem, že již není využito integrálního obrazu. Celý proces je popsán v [24].

7 IMPLEMENTACE

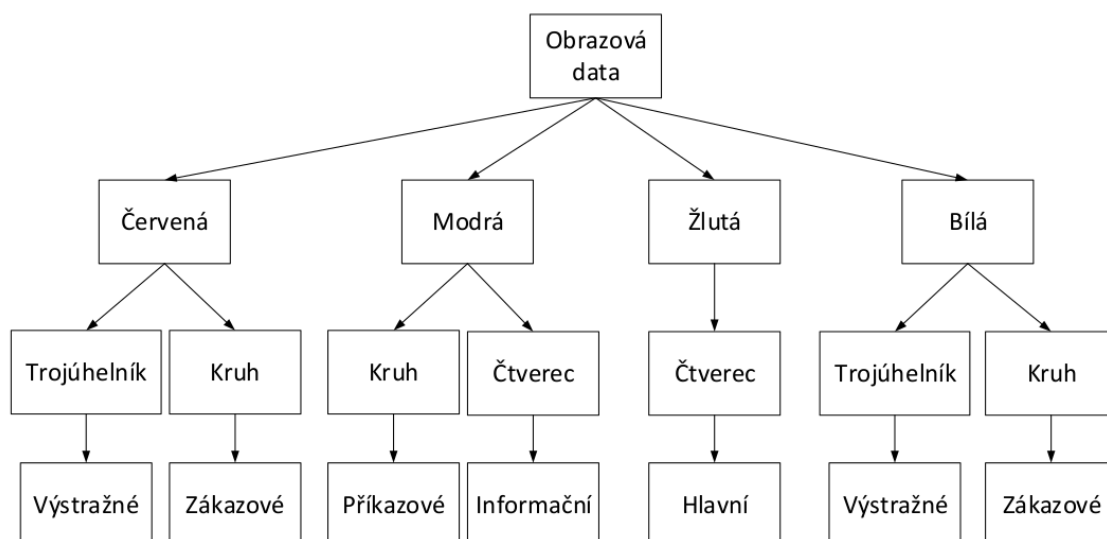
Algoritmus je implementován v prostředí Matlab verze R2014a. Algoritmus je tvořen celkem čtyřmi kroky. Prvním z těchto kroků je předzpracování obrazu, to nabízí možnost použití ekvalizace histogramu či lineárního zvýraznění kontrastu scény. Následuje segmentace, skládající se ze segmentace dle barvy, kde je obraz rozložen na pět binárních podobrazů (podobrazem je v textu myšlen binární obraz vzniklý barevnou segmentací), a segmentace dle velikosti, kde jsou odstraněny příliš malé či velké objekty v podobrazech. Poté je každý podobraz podroben identifikaci oblastí a pomocí radiometrických deskriptorů či Houghovy transformace jsou konkrétní objekty klasifikovány do příslušných skupin. Nakonec je každý objekt porovnán se vzory z konkrétní databáze pomocí techniky template matching nebo metody SURF. Výběr konkrétní databáze pro testování podobnosti objektů je vyznačen na obrázku 7.2. Obrázek 7.1 dokládá strukturu celého algoritmu.



Obr. 7.1: Schéma algoritmu.

7.1 Rozhodovací strom

Rozhodovací strom je vytvořen pomocí intuitivně navržených klasifikátorů. Těmito klasifikátory jsou barva, tvar apod. a jsou popsány v kapitole 7.3. Celý rozhodovací strom je znázorněn na obrázku 7.2. Počáteční snímek je dělen dle klasifikátoru barvy do čtyř mateřských uzlů (podobrazů), které jsou dále děleny, dle klasifikátoru tvaru, na jeden až tři mateřské uzly. Třetí uzel na obrázku 7.2 není pro jednoduchost zahrnut. Jedná se pouze o rozložení uzlu čtverec na obdélník a čtverec. Koncové uzly stromu zde nejsou listy, ale protože je finální metoda určení konkrétní značky vytvořena pomocí porovnání významných bodů objektu s databází vzorových značek, jsou koncovými uzly konkrétní databáze, do kterých je následně přistupováno. Dá se ovšem konstatovat, že je žlutý diamant, který je po dokončení větvení zařazen do uzlu *Hlavní pozemní komunikace*, již listem, avšak vzhledem k možné nepřesnosti detekce je i tato značka porovnávána se vzorem. Pátý podobraz spojuje vyřazené objekty z modrého a červeného podobrazu a není do obrázku 7.2 zahrnut.



Obr. 7.2: Rozhodovací strom.

7.2 Předzpracování

Předzpracování vstupního obrazu zahrnuje, vzhledem k rozmanité složitosti scény, pouze jasové transformace. Jedná se o lineární zvýraznění kontrastu scény, které je provedenou volanou funkcí *dynamic.m*, jejíž popis je uveden níže. Další možností je využít předdefinované funkce pro ekvalizaci histogramu *histeq.m*, nebo jasové transformace neprovádět.

function Jas = dynamic (Jas),

kde vstupní hodnotou je matice jasových složek zvoleného obrazu. V algoritmu je k tomuto primárně využíván první kanál prostoru YCbCr. Ve funkci je nejprve nalezena minimální a poté maximální hodnota jasu v konkrétním snímku. Poté jsou pomocí rovnice 4.1 vypočteny nové hodnoty jasu pro každý pixel obrazu.

Dalšími metodami, které je možné do jisté míry ještě považovat za předzpracování, nicméně svou podstatou spadají spíše k metodám segmentace, jsou filtry matematické morfologie 4.5. Ty jsou použity na již segmentované binární podobrazy. Využito je zejména otevření a uzavření.

Vzhledem k vysokému výskytu červených značek umístěných nad sebou je na podobraz červených pixelů aplikováno pouze otevření, následně doplněné zaplněním oblasti. Na modré podobrazy je naopak aplikováno pouze uzavření doplněné zaplněním oblasti. Zejména kvůli nemalým problémům při segmentaci značek typu přechod pro chodce, kde je hranice mezi dolními rohy bílého trojúhelníka s pikto-gramem a modrým čtvercem tento obklopující opravdu malý prostor modré bravy. Bílý podobraz je určen pouze jako doplňkový a proto na něj není aplikován, kromě zaplnění oblastí, žádný z filtrů. Podobně je tomu u žlutého podobrazu, který ve své podstatě slouží k detekci pouze jedné značky, navíc je vzhledem ke svým vlastnostem značka dokonale oddělena od pozadí bílými okraji.

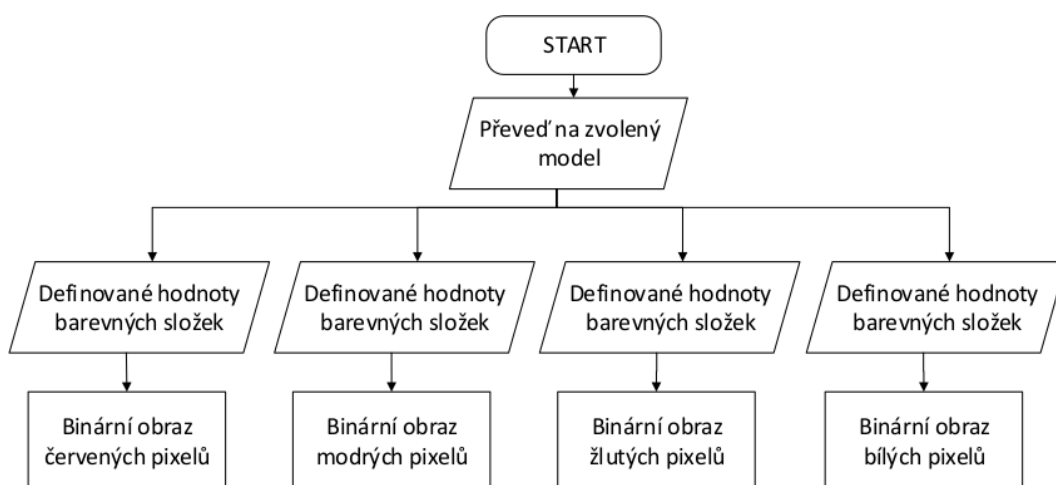
V průběhu práce na algoritmu bylo provedeno testování dalšího možného předzpracování, a sice zvýraznění hran v obraze za účelem efektivnějšího oddělení objektů od pozadí, nicméně tato metoda nebyla příliš efektivní.

7.3 Segmentace

V programu je využito hned dvou segmentací. První s ohledem na barvu objektu má na výstupu bíle (hodnota 1) pouze ty pixely spadající do tříd červené, modré, žluté či bílé barvy. Meze barev jsou určeny experimentálně a jsou definovány hodnotami složek obrazu ve volitelných barevných prostorech HSV, RGB nebo YCbCr. Zde je tedy využito známých barevných vlastností svislého dopravního značení. Jak již bylo uvedeno, barevná segmentace má za následek vytvoření čtyř binárních obrazů (podobrazů). Hodnota jedna v konkrétním podobraze značí přítomnost hledané barvy (barvy, odstínu a jasu), hodnota nula zastupuje zbylé nevyhovující body. Funkce vypadá následovně:

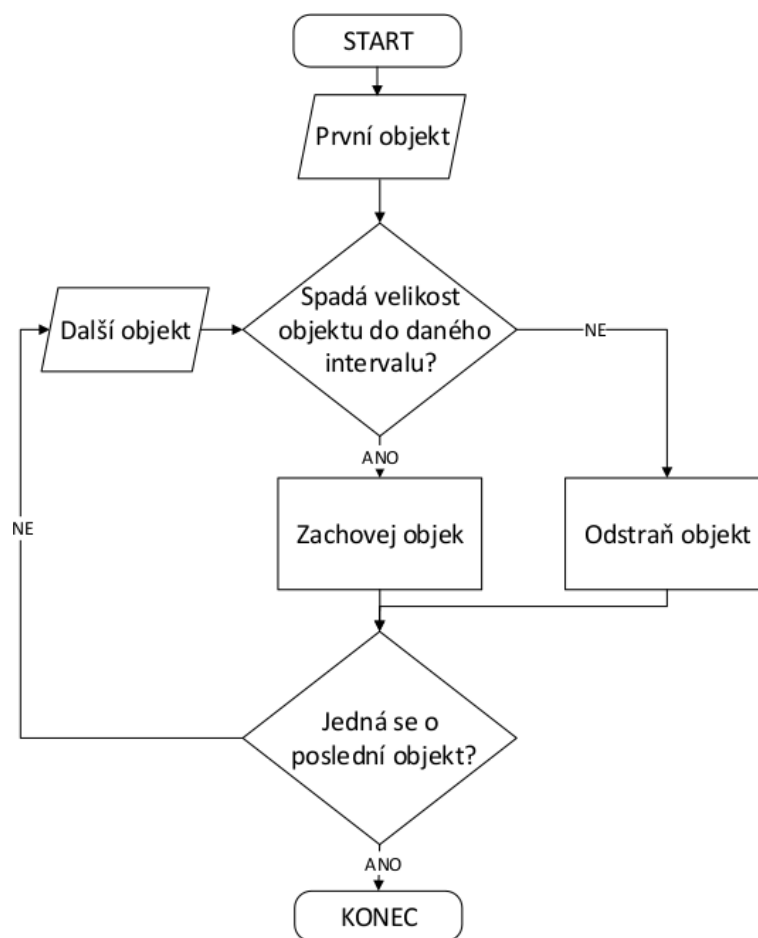
```
function [ RedIndex, BlueIndex, WhiteIndex, YellowIndex ]...  
= prostor( obr, typ_prostoru ).
```

Vstupem je zvolený snímek, další hodnotou je název (řetězec) zvoleného barevného prostoru, ve kterém chceme barvy hledat. Funkce provede segmentaci ve zvoleném prostoru a na výstup odešle binární obrazy (červený, modrý, bílý a žlutý). Celý proces barevné segmentace je naznačen následujícím diagramem 7.3.



Obr. 7.3: Vývojový diagram procesu segmentace s ohledem na barvu.

Druhá segmentace se provádí následně pro každý binární obraz, konkrétní barvy, zvlášť a definuje velikost objektů v intervalu zvolených prahových hodnot. Takto docílíme smazání malých, nepodstatných objektů, které by byly pro aplikaci template matchingu nestabilní. A také objektů příliš velkých. Objekty moc malé a objekty příliš velké, než aby měly co do činění s hledaným objektem, jsou v tomto kroku vymazány, respektive připojeny k pozadí (je jim přiřazena hodnota 0). K tomu je využito podmínek *if*, kterým však předchází identifikace oblastí, provedená pomocí předdefinované funkce *bwlabel*. Takto segmentované obrazy jsou poté připraveny k popisu objektů v nich se vyskytujících, více viz 7.4. Výsledkem segmentace uvedeně na obrázku 7.4 je binární obraz složený z objektů o počtu pixelů, kterým spadá do daného intervalu. 7.4.



Obr. 7.4: Vývojový diagram procesu segmentace s ohledem na velikost.

7.4 Popis objektů

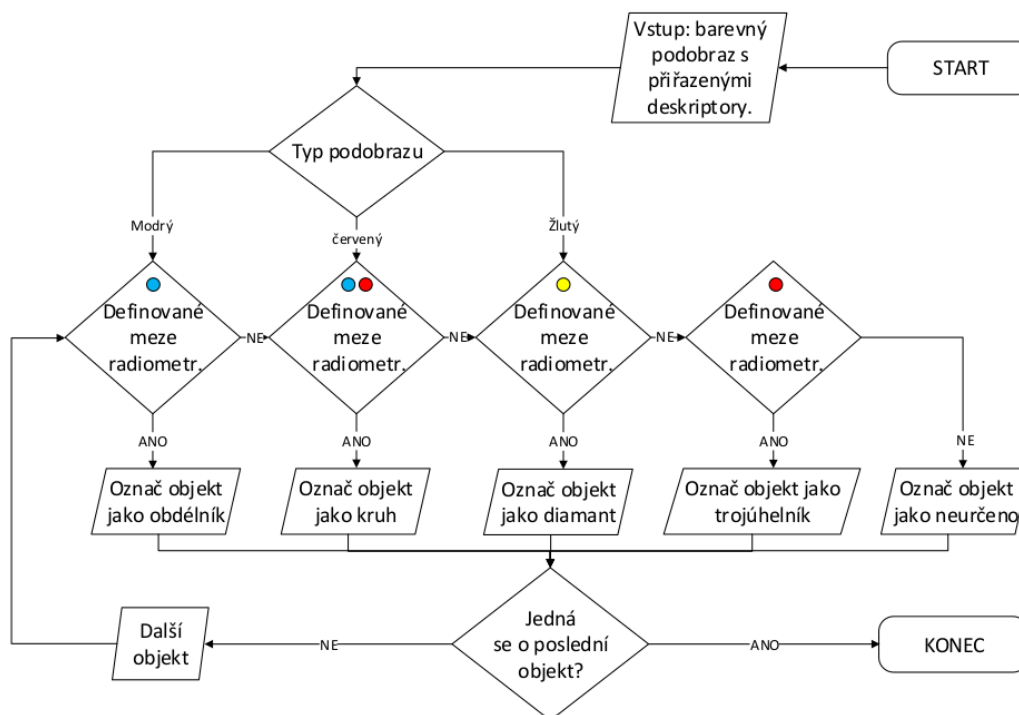
Po rozložení obrazu na čtyři již zmíněné podobrazy, kde každý nese informaci jedné konkrétní barvy scény, a provedení segmentace s ohledem na velikost, jsou na vstupu volané funkce tři hlavní podobrazy červené, modré a žluté barvy a jeden doplňkový bílé barvy, v binárním vyjádření. Následně je pomocí funkce *regionprops* proveden popis objektů pro každý hlavní obraz zvlášť. Funkcí *regionprops* je zjištěn obsah objektu *Area*, dále těžiště *Centroid*, délka hlavní osy objektu *MajorAxisLength*, ekvivalentní průměr objektu *EquivDiameter*, výstřednost *Eccentricity*, konvexnost *Solidity* a nakonec obdélník, ve kterém se objekt nachází *BoundingBox*. Následně je ještě dopočtena hodnota kompaktnosti pro každý objekt a hodnota poměru objektu ke svému orámování (*BoundingBox*). Tyto parametry následně používáme k dalšímu třídění daného objektu.

V části programu vyhrazené pro popis objektů je dále uměle vytvořen poslední podobraz logickým součtem obrazů červených a modrých pixelů. Tento podobraz je složen pouze z objektů, které nebyly klasifikovány ani v jednom podobrazu jako hledané tvary, toto je z důvodu, aby nedocházelo ke vstupu jednoho objektu do funkce pro rozpoznání dvakrát. Takto složený podobraz je vytvořen zejména kvůli značkám typu konec příkazu, které jsou kompletně přeškrtnuty červenou čarou a tedy rozděleny na dva půlkruhy, čímž se objekt stává neklasifikovatelným v podobruzích modré i červené barvy. Segmentace s ohledem na velikost je v algoritmu spolu s popisem objektů a klasifikací vsazena do jedné funkce nazvané *filtr*. Detailnější popis této funkce je uveden v 7.5.

7.5 Klasifikace radiometrickými deskriptory

Klasifikace segmentovaných objektů je provedena pomocí funkce *filtr.m*. Vstupem do této funkce jsou logické podobrazy červené, modré, bílé a žluté barvy. Dále řetězec definující typ detekce a nakonec samotný obrázek zvolené scény. Typ detekce určuje řízení funkce, ta se může řídit buď radiometrickými veličinami, nebo Houghovou transformací. Výstupem z funkce jsou struktury pro každý podobraz a podobrazy s odstraněnými objekty, které nespádají do intervalu povolené velikosti objektu. Ve funkci také vzniká uměle vytvořený modročervený podobraz, který spojuje nedetekované objekty z modrého a červeného podobrazu. To umožňuje následnou detekci značek, které nelze vzhledem k jejich vlastnostem správně klasifikovat ani v jednom ze zmíněných podobrazů.

Nejprve jsou odstraněny všechny nekonvexní objekty. Následně je s výhodou využito známých vlastností dopravního značení. Víme, že pro každou barvu existují pouze některé tvary dopravních značek, viz 7.2, například pro červenou barvu existuje pouze kruh (značky zákazové) a trojúhelník (značky výstražné). Pro modré značky je pak zbytečné hledat v obraze modrý trojúhelník. Z důvodu, že svislé značky se na obrázcích nenacházejí vždy pravoúhle promítnuty, nastává problém přesně definovat dané tvary pouze za pomoci poměru k čtyřúhelníku, ve kterém jsou vepsány. Největší problém nastává u modře zbarvených značek, proto je navíc u všech podobrazů využito dalších deskriptorů, které nám klasifikaci zefektivňují. Pro klasifikaci objektů z červeného podobrazu je pro kruh použito deskriptorů výstřednosti, kompaktnosti a poměru ekvivalentního poloměru k hlavní ose. Trojúhelník je pak určen intervalem poměru počtu pixelů trojúhelníka k počtu pixelů BoundingBoxu a hodnotou výstřednosti. Proces detekce tvaru za účelem klasifikace daného objektu je ve zjednodušené podobě popsán diagramem 7.5, kde barevné značky určují řízení směru rozhodování každého hlavního podobrazu (červený, modrý a žlutý podobraz).



Obr. 7.5: Vývojový diagram procesu detekce tvaru.

Po dokončení dělení je vytvořena struktura pro každý podobraz zvlášť, která obsahuje:

- přiřazené číslo značící konkrétní tvar;
- souřadnice, kde se značka v obraze nachází;
- název konkrétního tvaru;
- jeho obsah;
- polohu těžiště a další.

7.6 Klasifikace Houghovou transformací

Za předpokladu, že je zvolena klasifikace pomocí této metody, provede algoritmus nejprve popis objektů pro každý podobraz (vzhledem k časové náročnosti a předpokládané přesnosti Houghovy transformace jsou zde zpracovávány pouze podobrazy červené, modré a žluté barvy). Následně jsou pomocí funkce *regionprops* zjištěny nejdůležitější deskriptory každého objektu jako je těžiště, ohraničení oblasti, konvexnost a podobně. Poté jsou provedeny následující kroky:

1. Vstupní snímek je převeden na jasové hodnoty.
2. Je provedeno vystřížení konvexního objektu dle jeho ohraničení.
3. Objekt s nevyhovujícím stupněm konvexnosti je označen jako *neurčeno*.
4. Vystřížený objekt je vyhranován pomocí Cannyho detektoru.
5. Objekt vstupuje do funkce *houghFCN* jako binární obraz.

První vstupní proměnnou funkce *houghFCN* je číslice udávající typ barevného podobrazu (např. 1 pro červený podobraz), dále binární obraz hran podezřelé oblasti, struktura obsahující údaje o konkrétním podobrazu, pořadové číslo daného objektu ve struktuře a počet pixelů, o který je podezřelá oblast rozšířena. Rozšíření je zavedeno z důvodu, že je daný objekt klasifikován pomocí hran, které běžně náleží přesně souřadnicím ohraničení oblasti, a proto je nutné tuto oblast rozšířit o malý počet pixelů. Výstupem v případě pozitivní detekce je jedna struktura pro nalezený tvar obsahující údaj o jeho ohraničení, v tomto kroku zmenšeném o hodnotu předchozího rozšíření, a těžišti v celém obraze.

Funkce *houghFCN* je rozdělena do dvou částí. V první části je provedena detekce kruhu a trojúhelníka přesně v tomto pořadí. Pokud je objekt identifikován v prvním kroku jako kruh (s pomocí funkce *imfindcircles*), je detekce tvaru ukončena. V opačném případě je provedeno testování pro trojúhelník. Zde je využito funkce *houghpeaks*

a *houghlines* ke konstrukci úseček. Následně jsou vypočteny všechny možné kombinace úseček (kombinace bez opakování), které by mohly teoreticky tvořit trojúhelník. Tyto kombinace úseček jsou následně testovány, zdali skutečně tvoří trojúhelník. K tomu je využito obecných geometrických vztahů pro konstrukci trojúhelníka. První část této funkce je určena výhradně pro podobraz červených objektů.

Druhá část je pak určena k detekci objektů z modrého a žlutého podobrazu. Zde je opět využito funkce *imfindcircles* nebo vztahů pro konstrukci čtyřúhelníku. Výstupem jsou vždy tři proměnné značící konkrétní identifikovaný tvar.

Za předpokladu, že tvar není klasifikován ani v jednom kroku této funkce, jsou všechny výstupní proměnné rovny nule. Pokud je objekt identifikován jako určitý tvar, je proměnná pro tento tvar strukturou nesoucí údaje o poloze těžiště daného objektu a jeho ohraničení v celém obraze. Ostatní proměnné jsou rovny nule. Následně je ve funkci *filtr* vytvořena struktura pro konkrétní podobraz podobně jako v 7.5.

Nutno poznamenat, že celý proces je velice časově náročný. Už z důvodu, že pokud není objekt identifikován jako kruh, je provedena vždy další detekce na trojúhelník či čtverec, což celý proces značně zpožďuje. Dalším problémem je nedokonalost či nespojitost, úseček tvořících konkrétní tvar. Tento problém lze pozorovat hlavně při detekci čtverce, která je dosti závislá na kvalitě snímku a kvalitě následného vyhranování. Vývoj takto navržené metody detekce tvaru, která je založena na Houghově transformaci, byl značně zavádějící. Algoritmus byl vyvíjen na snímcích obsahujících pouze geometrické entity a bílé pozadí, což celý proces detekce značně urychlovalo.

Nakonec bylo nutné zapojit před vstup do funkce *houghFCN* podmínku, která vylučuje nekonvexní objekty. Bezpochyby rychlejším způsobem je použít jako vstup do této funkce vyhranovaný logický obraz daného objektu. To však znamená, že proces detekce tvaru bude, podobně jako u radiometrických deskriptorů, závislý na přesné segmentaci.

7.7 Rozpoznání

Již známe barvu a tvar konkrétního objektu, provedeme tedy vystřižení objektu ze vstupního obrazu a pomocí funkce *detectSURFFeatures* nebo *template matchingu* porovnáme objekt s konkrétními vzory, nebo významnými body. Pro tento účel jsou připraveny databáze, které jsou popsány v sekci 7.8. Výsledkem rozpoznání je vzor, který nejvíce odpovídá detekované značce a svou podobností přesahuje hodnotu zvoleného prahu, který byl určen empiricky pro každou databázi zvlášť.

7.7.1 Template matching

Implementace této metody je založena na srovnání podezřelé oblasti, tj. oblasti nesoucí znaky barvy a tvaru konkrétního typu značení, se všemi šablonami z konkrétní databáze pomocí prosté 2D korelace 6.1.1. Databáze je zvolena v závislosti na tvaru a barevných vlastnostech konkrétní podezřelé oblasti.

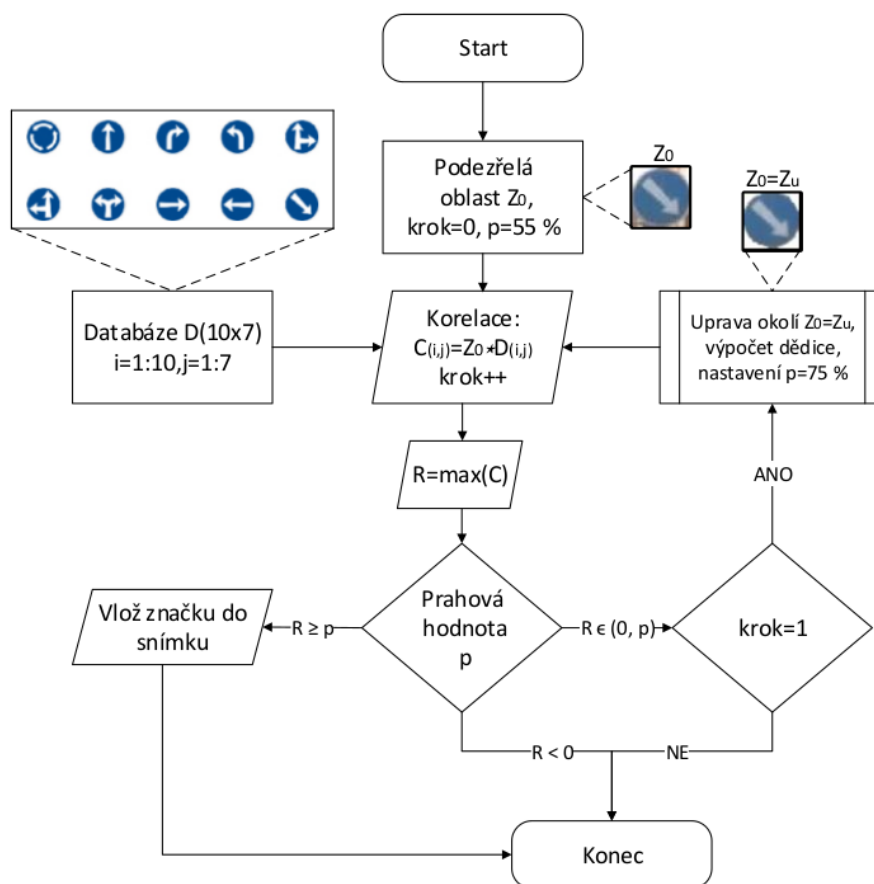
Nejprve je vybraná oblast normalizována na velikost vzorů z konkrétní databáze. Tato normalizace velikosti značně snižuje vliv zkosení na výsledky identifikace. Korelace se provádí cyklicky pro každý vzor z databáze, který je navíc reprezentován sedmi snímky pootočenými vždy o tři stupně. Výsledkem vzájemné korelace podezřelé oblasti a jedné konkrétní šablony je tedy sedm korelačních koeficientů. Pokud obsahuje databáze například 5 značek, bude výsledná matice korelačních koeficientů nabývat rozměru 5×7 čili pět řádků, které reprezentují pětici šablon a sedm sloupců, které představují konkrétní natočení šablony. Z této matice je vybrán největší korelační koeficient. Ten je poté porovnán s prahovou hodnotou, nabývající cca. 70% možné shody, která představuje určitou míru věrohodnosti. Prahová hodnota věrohodnosti do jisté míry zamezuje výskytu falešných detekcí. Pokud koeficient přesahuje tento práh, je rozhodnuto, že se jedná o konkrétní značku. Ta je následně, dle řádku, ve kterém se koeficient vyskytuje, vykreslena v obraze scény a podezřelý (již už identifikovaný) objekt je orámován.

Pokud je maximální koeficient záporný čili menší než 50% možné shody, je porovnávání ukončeno. Za předpokladu, že je daný koeficient kladný a menší nežli prahová hodnota, výsledná matice koeficientů je uložena do proměnné *dedic*. Tato proměnná je následně přepočtena, tak aby nejvyšší koeficient nabýval hodnoty 55% možné shody. Toho je docíleno výpočtem:

$$\mathbf{D}_{(x,y)} = 0,1 \frac{\mathbf{D}_{\mathbf{0}(x,y)}}{\mathbf{D}_{\mathbf{0max}}}, \quad (7.1)$$

kde $\mathbf{D}_{(x,y)}$ je výsledná hodnota koeficientu dané značky konkrétní rotace, $\mathbf{D}_{\mathbf{0}(x,y)}$ je původní koeficient dané značky konkrétní rotace a $\mathbf{D}_{\mathbf{0max}}$ je hodnota maximálního koeficientu výsledku korelace. Následně je provedeno odstranění pozadí podezřelého objektu pomocí funkce *Warea*, tak aby byla možná značka co nejpodobnější šabloně. Je tedy provedeno přebarvení okolí objektu na bílo. Následně je opět provedena korelace. Nakonec je matice nových koeficientů sečtena s normovanou maticí dědičnosti. Opět je vybrán nejvyšší koeficient, který je porovnán s prahovou hodnotou. Práh tentokrát odpovídá hodnotě cca 75% maximální možné shody. Pokud ani tentokrát není shoda větší nežli zvolený práh, je rozhodnuto, že daný objekt není značkou.

Tento proces zajišťuje funkce *template.m*, kde jsou vstupními hodnotami: struktura konkrétního barevného podobrazu, obsahující údaje o objektech, které jsou popsány v části 7.5. Dále databáze konkrétních značek, snímek scény, údaj o barvě, koeficient k určující danou podezřelou oblast (řádek ve struktuře podobrazu). Následně koeficient a určující pořadí hledané značky (pokud ještě nebyla nalezena žádná značka, je a rovno jedné) a nakonec parametr pr , který určuje polovinu počtu pixelů, o které se má porovnávaná oblast zvětšit (běžně je tento parametr nulový, pokud však dojde k prohledávání bílého podobrazu, je tento parametr nastaven na rozdíl výšky celé značky a výšky nalezené oblasti). Výstupem funkce je obrázek scény s nejshodnější šablonou, která je umístěna v levém dolním rohu, a orámovaný objekt (v případě nálezů značky) s kódem značky. Druhou výstupní proměnnou je v případě nálezů značky inkrementovaná hodnota a . Obrázek 7.6 ukazuje proces identifikace oblasti pomocí této funkce.



Obr. 7.6: Vývojový diagram algoritmu *template.m*.

7.7.2 SURF

Metoda SURF je implementována ve funkci *surfmatch.m*. Vstupními hodnotami jsou: struktura nesoucí informace o objektech konkrétního podobrazu, databáze značek, která je vybrána podle typu objektu, snímek dané scény, informace o barvě podobrazu, číslo pořadí testovaného objektu ve struktuře a koeficient *a*, který udává pořadí hledané značky stejně jako v případě funkce *template.m*.

```
function [output2, a] = surfmatch(barva, typ, output, output2, color, k, a)
```

Výstupy z této funkce jsou taktéž stejné jako v předchozím případě. Funkce je založena na podobném principu jako template matching, avšak již nejsou porovnávány hodnoty jasů nebo jedné z barevných složek vzoru a šablony, ale shoda v jejich významných bodech. Celý proces je opět nepatrně odlišný pro některé databáze. Velkou roli zde hrají značky bez piktogramu, které mají velice málo významných bodů. Vzhledem k jejich vzhledu lze však tento jev předpokládat.

Předpokladem pro provedení identifikace pomocí této funkce, je mimo jiné databáze značek obsahující významné body pro každou šablonu. Prvním krokem v této funkci je vystřížení podezřelé oblasti ze vstupního snímku. Ta je poté převedena do stupňů šedi a normalizována na velikost shodnou s velikostí šablony (toto se osvědčilo jako nejlepší způsob porovnávání a to i za předpokladu, že jsou body získané touto metodou nezávislé na změně rozměrů). Pokud se jedná o značku bez piktogramu, je automaticky vybrána struktura významných bodů se sníženou hodnotou (*MetricThreshold*) síly těchto bodů na 10%, to způsobuje nalezení většího množství bodů v každém obraze, a taktéž je vytvořena matice těchto bodů, získaných pomocí funkce *detectSURFFeatures*, pro testovaný objekt.

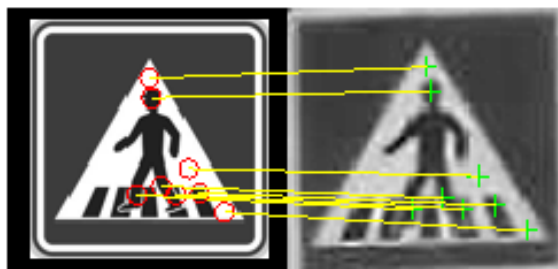
```
znacka_pts = detectSURFFeatures(znacka, 'MetricThreshold', strong);
```

kde proměnná *strong* určuje sílu nalezených bodů, zde tedy 100. Následně je vytvořeno pole *index_pairs*, které obsahuje shodné body každé šablony s objektem a další pole *metric*, obsahující vzájemnou euklidovskou vzdálenost těchto bodů.

```
[index_pairs{i}, metric{i}] = (matchFeatures(eval(['typ.', 'ref_features', ...  
num2str(strong), '(i)', 'ref_features']), znacka_features, 'MatchThreshold', p));
```

kde proměnná *p* značí procentuální odchylku od dokonalé shody, *p* je obvykle rovna 1. Následně je provedeno odstranění bodů s křížícími se spojnicemi z každé shody pomocí funkce *estimateGeometricTransform*, která využívá iterativní metody RANSAC (*RANdom Sample Consensus*).

Podle výsledku je vybrána značka s nejvyšším počtem bodů a nejnižšími průměrnými metrikami mezi podobnými body.



Obr. 7.7: Ukázka výsledných korespondujících bodů metody SURF.

V opačném případě, tedy jedná-li se o značku obsahující piktogram, program provede porovnávání významných bodů s hodnotou 100% síly. Pokud má kterákoli šablona alespoň šest společných bodů s testovaným objektem, je opět vybrána nejpodobnější značka. Pokud je nalezených bodů méně, je provedeno snížení síly konkrétních nalezených bodů o 30%. To se děje tak dlouho, dokud není nalezena alespoň jedna shoda dosahující minimálně šest společných bodů, nebo není dosaženo síly bodů 40%. Poté je opět proveden součet převrácených hodnot největšího počtu bodů, nejvyššího počtu lineárních bodů a hodnoty nejnižšího rozdílu vzdáleností těchto bodů. Podle nejnižší hodnoty tohoto součtu je vybraná šablona přiřazena objektu a objekt je identifikován jako tato šablona. Další kroky funkce jsou pak shodné s kroky ve funkci *template.m*, tedy zobrazení šablony a orámování značky s jejím kódovým označením.

7.8 Databáze značek

Databáze je jedním z nejdůležitějších elementů pro funkci užitých metod rozpoznání. Je velice důležité dané databáze naplnit obrazy šablon, které spojují určité vlastnosti pro tyto šablony typické. Proto je databází hned několik, a ne vždy obsahují pouze jeden typ daného značení. Příkladem může být databáze zákazových značek, která mimo značky zákazové obsahuje i některé značky upravující přednost. Tato databáze je navíc ještě rozdělena, podobně jako databáze značek výstražných, do dvou skupin, a sice značky obsahující piktogram a značky bez piktogramu. Výsledkem tohoto rozdělení je deset různých databází lišících se nejen tvarem a barevnými vlastnostmi značení ale také přítomností piktogramu.

Nejprve jsou načteny všechny obrázky obsahující konkrétní typ značek, toho je docíleno rozdělením těchto obrázků do složek. Následně je pro každou značku vytvořeno šest rotovaných kopií pomocí operace rotace, popsané v sekci 4.2.2. Rotace je provedena pro každou značku od -9° po 3° do 9° . Tímto je částečně docíleno

potlačení vlivu rotace značení na proces template matchingu. Rotované obrazy a i obraz konkrétní značky jsou následně obstříženy, tak aby byla šablona co nejpodobnější oblasti, která bude nalezena algoritmem. Pozadí značení je vyběleno a značky jsou poté uloženy do struktury, tak aby jejich řádek odpovídal konkrétní značce a sloupec její rotaci. Nakonec je do posledního sloupce struktury uložena originální šablona, která je při úspěšné identifikaci zobrazována v dolním rohu obrazu.

V dalším kroku je každá originální šablona převedena do stupňů šedi. Pomocí funkce *detectSURFFeatures* a *extractFeatures* jsou detekovány a uloženy významné body každé šablony. Ty jsou poté uloženy do struktury, a celý proces se opakuje pro další databázi značek.

I přes fakt, že databáze zdaleka neobsahují všechny dopravní značky, se kterými je možné se na pozemních komunikacích v České republice setkat, jsou všechny databáze dohromady naplněny 151 značkami. Mezi značky, které nejsou do databází zahrnuty, patří hlavně značky dodatkové, dále všechny informační značky zelené barvy (vzhledem k faktu že se na nich nejčastěji objevuje různý text). Dále pak značky obsahující číselné údaje, jako jsou například zákazové značky definující okamžitou hmotnost, výšku či šířku vozidla a podobně. Některé značky se vzhledem ke struktuře algoritmu objevují i ve dvou databázích současně, sem patří pouze značka zákaz zastavení a značka zákaz stání. Algoritmus je však ošetřen tak, aby nedocházelo ke dvojí identifikaci jedné značky zároveň.

Databáze vzorových značek byla pořízena s pomocí [25]. K vývoji algoritmu bylo použito především dokumentace společnosti Mathworks z [20].

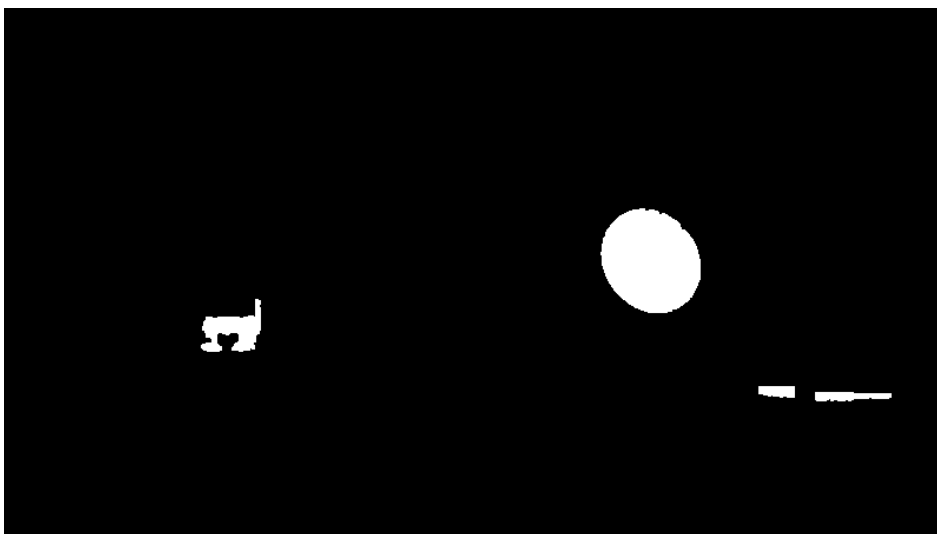
8 VÝSLEDKY ALGORITMU

V této kapitole budou vyobrazeny nejprve výsledky detekce a identifikace dopravních značek ve vybraných snímcích. Následně bude uvedena statistika úspěšnosti každé z metod, viz 8.1.

Výsledky jsou zobrazovány přímo v obraze, a sice v jeho spodní části. Identifikovaná značka je černě orámována, v oblasti je lineárně zvýšen jas a značce je přiřazen kódový název. První detekovaná značka je zobrazena vždy v levém dolním okraji snímku. V případě detekce a identifikace více značek je další značka zobrazena na pravo od předchozí, to lze pozorovat na obrázku 8.6.



Obr. 8.1: Ukázka správné identifikace pomocí metody SURF.



Obr. 8.2: Ukázka segmentace předchozí scény.



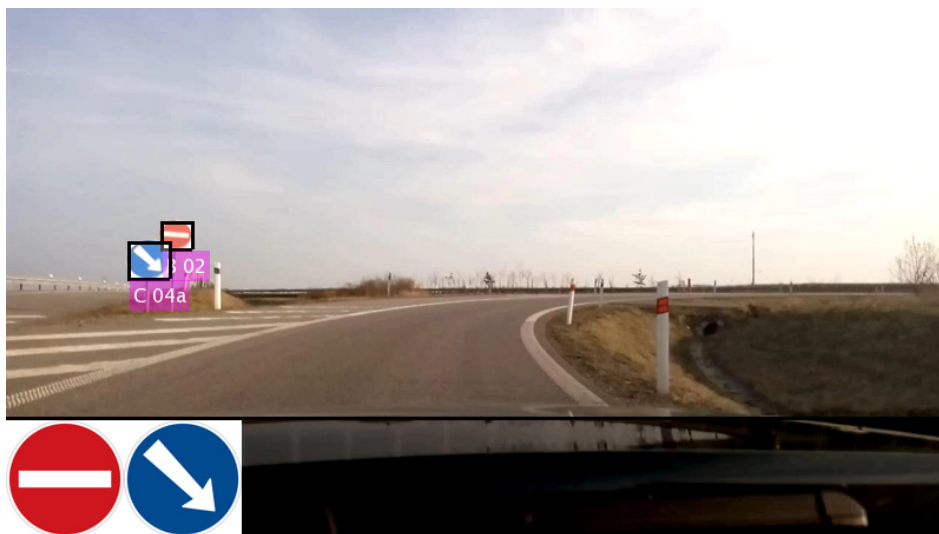
Obr. 8.3: Ukázka správné identifikace pomocí metody SURF.



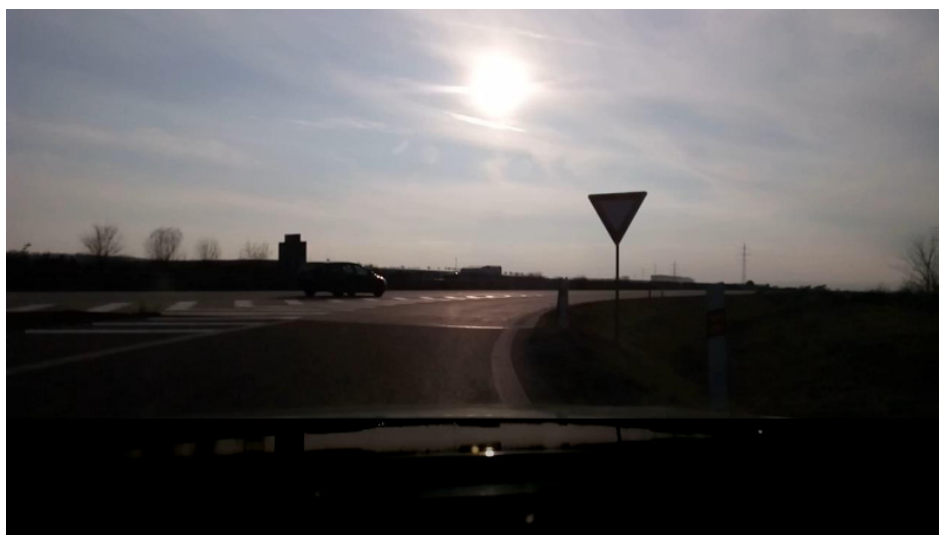
Obr. 8.4: Ukázka správné identifikace pomocí metody template matching.



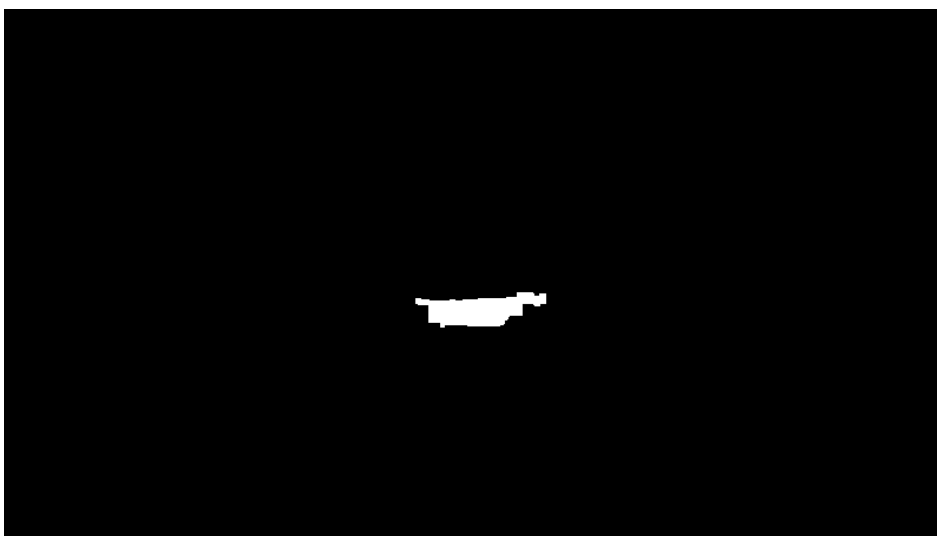
Obr. 8.5: Ukázka správné identifikace pomocí metody template matching.



Obr. 8.6: Ukázka správné detekce pomocí Houghovy transformace.



Obr. 8.7: Ukázka nulové detekce zapříčiněné nepříznivými světelnými podmínkami.



Obr. 8.8: Ukázka segmentace předchozí scény.

První obrázek 8.1 zobrazuje úspěšnou detekci a identifikaci pomocí metody SURF. Scéna je segmentována v prostoru YCbCr, klasifikace je zde založena na regionech a není provedeno žádné předzpracování snímku. Obrázek 8.2 ukazuje segmentaci dle červené barvy předchozího obrázku. Další obrázek 8.3 je předzpracován pomocí jasové transformace, je segmentován v HSV modelu a je taktéž klasifikován pomocí radiometrických deskriptorů. K identifikaci je zde opět využita metoda SURF.

Na dalších obrázcích je pak použita k identifikaci metoda založená na template matchingu. Obrázek 8.4 je předzpracován pomocí jasové transformace, segmentován v modelu YCbCr a klasifikován radiometrickými deskriptory. U obrázku 8.5 je patrné použití ekvalizace histogramu, segmentace je provedena v prostoru RGB. Klasifikace je opět provedena pomocí radiometrie. Dále je zde možné pozorovat menší ohraničení značky. Z toho je patrné, že segmentace červené barvy nenašla celistvý objekt, ten následně nebyl identifikován jako červený trojúhelník. V tomto případě byla značka tedy identifikována s pomocí bílého podobrazu, u kterého je proces identifikace naprosto stejný, avšak vystřižený objekt je zvětšen o hodnotu poměru velikosti značky s velikostí jejího bílého vnitřního prostoru, následně identifikován a ohraničen obdélníkem o původních rozměrech obdélníka detekovaného tvaru.

Poslední obrázek úspěšné detekce a identifikace značení 8.6, je segmentován v HSV prostoru a klasifikován s pomocí Houghovy transformace.

Na posledním obrázku scény 8.7 je značka ztracena již v prvním kroku, a sice při segmentaci. Toto dokládá obrázek 8.8, který ukazuje výsledek segmentace dle červené barvy v obraze. Důvodem špatné segmentace a následné nulové detekce a identifikace objektu je špatné osvětlení scény.

8.1 Statistické vyhodnocení úspěšnosti

Měření úspěšnosti proběhlo na notebooku s procesorem Intel® Celeron®, 1.83 GHz. V této sekci bude uvedena úspěšnost jednotlivých metod. Protože je možných kombinací skutečně mnoho, bude úspěšnost každé kombinace předzpracování, segmentace v barevném prostoru, klasifikace a identifikace, sestavena vždy na dvaceti vybraných snímcích. Nakonec bude zvolena metoda s nejlepšími výsledky, která bude otestována na celé škále pořízených snímků. Následující tabulka vysvětluje použité zkratky při měření a vyhodnocování úspěšnosti jednotlivých metod:

Označení	Předzpracování	Prostor	Detekce	Identifikace
Z_H_R_T	žádné	HSV	radiometrie	template matching
J_H_R_T	jasová tr.	HSV	radiometrie	template matching
E_H_R_T	ekvalizace	HSV	radiometrie	template matching
Z_R_R_T	žádné	RGB	radiometrie	template matching
J_R_R_T	jasová tr.	RGB	radiometrie	template matching
E_R_R_T	ekvalizace	RGB	radiometrie	template matching
Z_Y_R_T	žádné	YCbCr	radiometrie	template matching
J_Y_R_T	jasová tr.	YCbCr	radiometrie	template matching
E_Y_R_T	ekvalizace	YCbCr	radiometrie	template matching
Z_H_R_S	žádné	HSV	radiometrie	metoda SURF
Z_H_H_S	žádné	HSV	Hough. tr.	metoda SURF

Tab. 8.1: Tabulka zkratek možných kombinací metod.

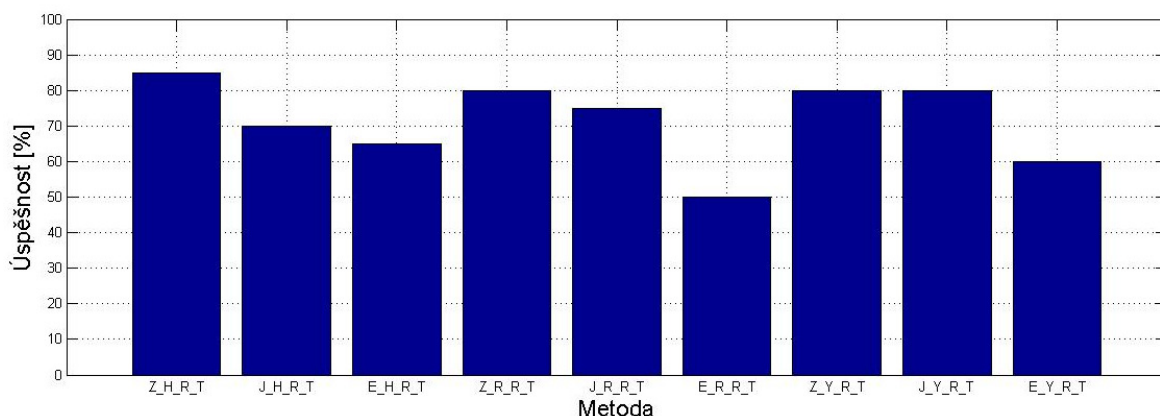
Další zkratka na třetí pozici značí použití Houghovy transformace. Na čtvrté pozici se poté vyskytuje buď písmeno T jako template matching, nebo S, které značí užití metody SURF k výsledné identifikaci. Výše uvedená tabulka tedy neobsahuje všechny možné kombinace metod, těch je celkem 36.

Za předpokladu, že identifikace neproběhla úspěšně nebo byl-li v obraze, mimo úspěšné identifikace, detekován a identifikován i jiný objekt, který není značkou, byly

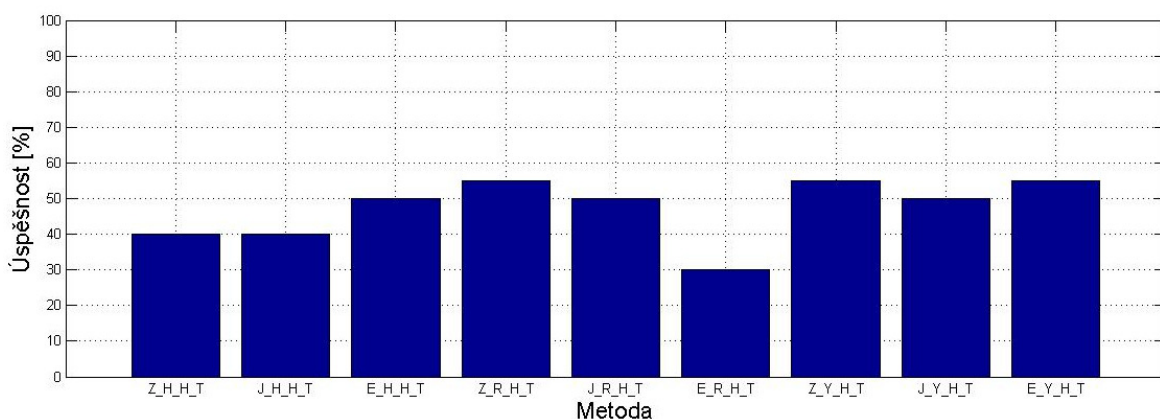
tyto výsledky považovány automaticky za chybné. Procentuální úspěšnost každé z metod tedy zahrnuje pouze stav, kdy byl v obraze detekován a následně identifikován přesný typ a počet značek, které se v něm skutečně nacházely. Následující tabulky ukazují výsledné procentuální úspěšnosti jednotlivých metod. Měření proběhlo celkem na dvaceti značkách. Výsledné časy jednotlivých kombinací metod odpovídají průměrnému času vypočtenému z pozitivních i negativních identifikací.

Metody	úspěšnost [%]	celkový čas [s]
Z_H_R_T	85	5,176
J_H_R_T	70	4,945
E_H_R_T	65	5,570
Z_R_R_T	80	5,302
J_R_R_T	75	4,838
E_R_R_T	50	4,772
Z_Y_R_T	80	5,491
J_Y_R_T	80	5,432
E_Y_R_T	60	4,154
Z_H_H_T	40	8,121
J_H_H_T	40	6,915
E_H_H_T	50	5,506
Z_R_H_T	55	5,552
J_R_H_T	50	6,081
E_R_H_T	30	3,484
Z_Y_H_T	55	6,868
J_Y_H_T	50	7,234
E_Y_H_T	55	4,983

Tab. 8.2: Tabulka úspěšnosti identifikace metodou template matching.



Obr. 8.9: Detekce radiometrickými deskriptory.



Obr. 8.10: Detekce Houghovou transformací.

Výše uvedené grafy ukazují úspěšnosti možných metod předzpracování, segmentace a detekce, pro identifikaci metodou template matching. Graf 8.9 udává úspěšnost algoritmu pro všechny možné segmentace. Detekce objektů a jejich klasifikace je zde provedena v závislosti na přidělených radiometrických deskriptorech. Nejúspěšnější je zde segmentace v HSV prostoru a to bez provedení předzpracování obrazu. Průměrná doba celého procesu je zde 5,2 s.

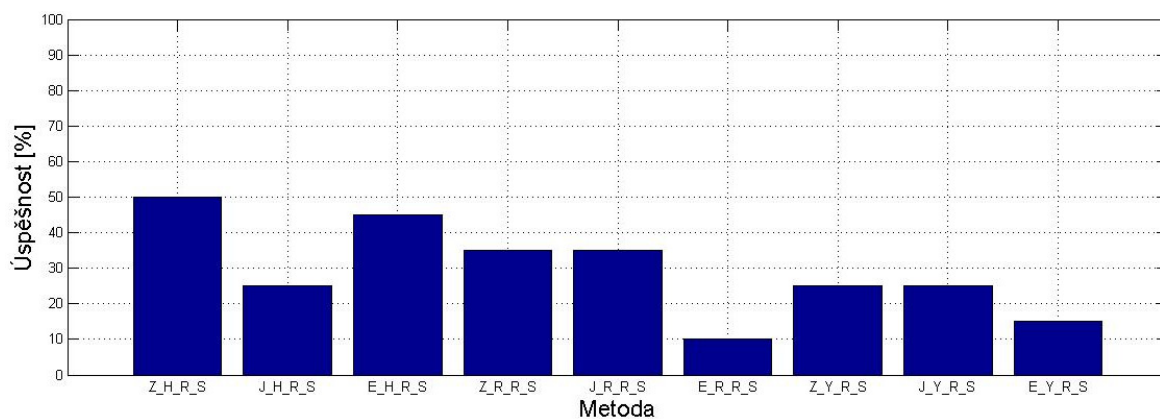
Graf 8.10 udává úspěšnost algoritmu opět pro segmentace všech barevných prostorů. Klasifikace objektů je zde provedena v závislosti na výsledcích Houghovy transformace. Nejúspěšnější metodou je v tomto případě segmentace v prostorech RGB a YCbCr, opět bez předzpracování obrazu. Průměrný čas obou metod se pohybuje v rozmezí cca. 5–5,6 s. Vzhledem k problému v Houghově transformaci detekovat čtverec či obdélník je statistika tímto faktorem dosti ovlivněna, v testovací skupině je celkem 6 čtvercových značek, které metoda není schopna detekovat.

Tabulka úspěšnosti identifikace pomocí metody SURF je vyobrazena níže. Důvodem nízké úspěšnosti této metody je do jisté míry neschopnost touto metodou identifikovat značky bez piktogramu. Metoda vzhledem ke své nezávislosti na rotaci často chybuje a zaměňuje značky se zrcadlovými, hlavně v případě příkazových značek.

Metody	úspěšnost [%]	celkový čas [s]
Z_H_R_S	50	4,309
J_H_R_S	25	4,034
E_H_R_S	45	3,982
Z_R_R_S	35	3,616
J_R_R_S	35	3,820
E_R_R_S	10	3,753
Z_Y_R_S	25	3,657
J_Y_R_S	25	3,924
E_Y_R_S	15	3,575

Tab. 8.3: Tabulka úspěšnosti identifikace metodou SURF.

Graf 8.11 ukazuje úspěšnost metod, které svou detekci a klasifikaci zakládají na radiometrických deskriptorech. Identifikace je zde provedena pomocí metody SURF. Nejúspěšnější z metod je metoda Z_H_R_S, tedy metoda bez předzpracování vstupního snímku a segmentace je provedena v HSV prostoru. Celkový čas činí 4,309 s.

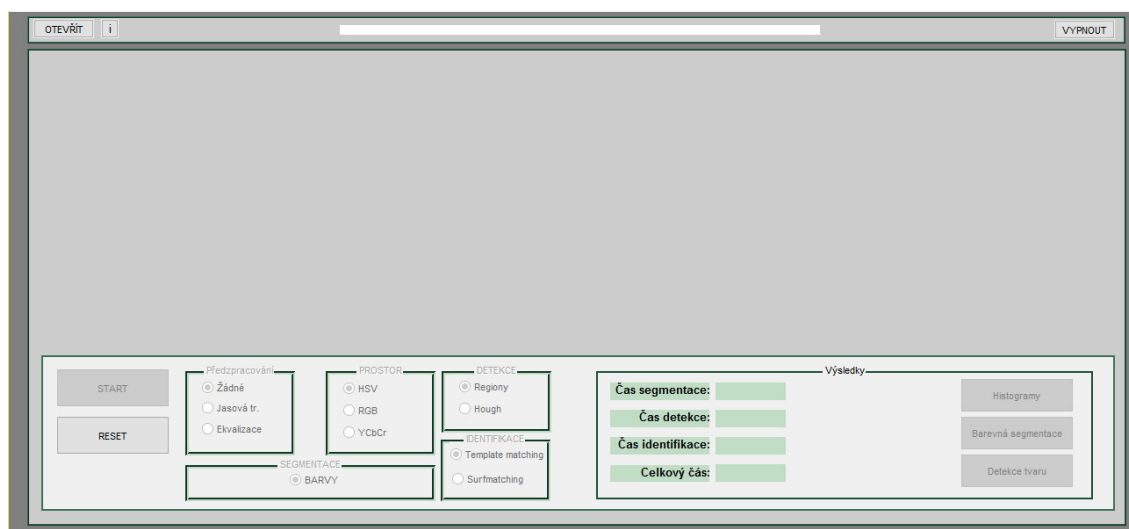


Obr. 8.11: Detekce radiometrickými deskriptory.

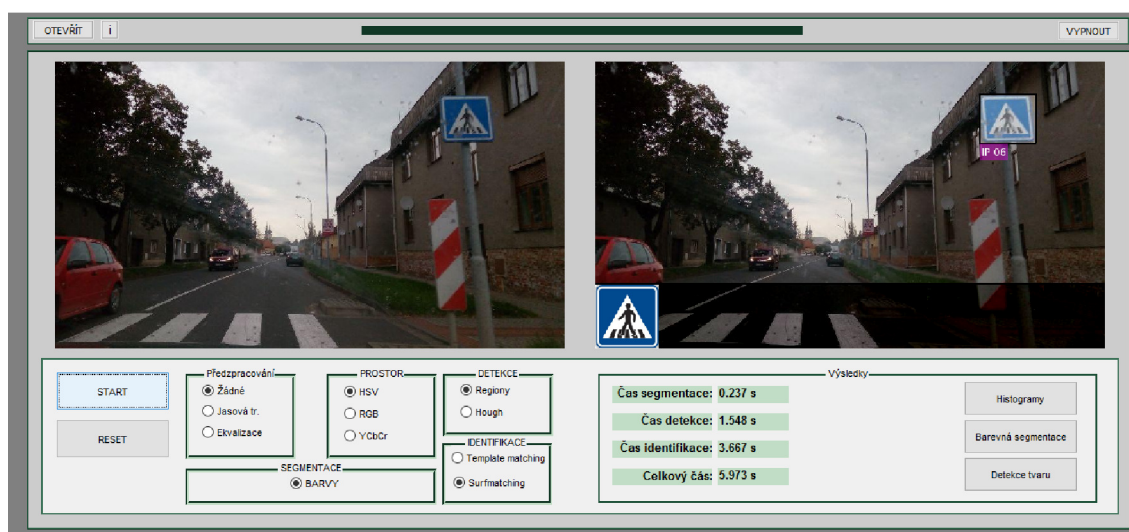
Poslední kombinací metod je pak kombinace detekce založené na Houghově transformaci a identifikace pomocí metody SURF. Vzhledem k dosavadním výsledkům obou ze jmenovaných je však zcela jisté, že úspěšnost této kombinace bude velice nízká. To hlavně z důvodu, že Houghova transformace má problémy s detekcí čtverce, či obdélníka a metoda SURF má problémy se značkami bez piktogramu. Vzhledem ke statistice výsledků by bylo nutné sestavit testovací soubor, který by neobsahoval tyto značky, statistika by však tímto byla ovlivněna a výsledky by byly nesrovnatelné s předchozími metodami.

Výsledná úspěšnost nejlepší kombinace, tedy Z_H_R_T, testované na všech 164 pořízených snímcích, skýtajících okolo 200 dopravních značek, byla 46,94%. Některé snímky však obsahují značky, které v databázích nejsou zavedeny (v takovém případě byla jako úspěšná identifikace považována pouze ta, kdy značka, která není v databázi, nebyla identifikována), to úspěšnost dosti zkresluje. Druhým faktorem ovlivňujícím tuto úspěšnost je kvalita některých fotografií, které byly často foceny za špatných světelných podmínek, špatného počasí a při vyšší rychlosti automobilu, což způsobovalo rozmazání snímků. Průměrný čas této metody, vypočtený na množině 164 snímků, byl 5,5 s (Pro srovnání: na vypůjčeném notebooku s procesorem Intel Core i5-3230M, 2.6 GHz byl průměrný čas této kombinace metod 1,5 s).

9 UŽIVATELSKÉ PROSTŘEDÍ



Obr. 9.1: Uživatelské prostředí po spuštění.



Obr. 9.2: Uživatelské prostředí po dokončení identifikace.

Uživatelské prostředí se spouští ze souboru *Identifikace.m*. Po startu programu se zobrazí úvodní obrazovka 9.1. Všechny v tomto kroku neproveditelné operace jsou uzamčeny a je možné pouze program vypnout, restartovat nebo kliknout na ikonu *i*, která zobrazí zprávu obsahující návod k použití programu, či použít tlačítko *OTEVŘÍT*.

Prvním krokem je stisknutí tlačítka *OTEVŘÍT*, které umožní vybrat snímek k testování. Po výběru snímku je možné opět kliknout na ikonu *i*, která zobrazí zprávu obsahující návod k dalšímu kroku v programu. Je možné zvolit libovolnou kombinaci předzpracování obrazu, barevný prostor k segmentaci, metodu detekce a identifikace objektu, nebo nastavení ponechat.

Po nastavení dané kombinace je možné stisknout tlačítko *START*, které spustí proces rozpoznání značení ve zvoleném obraze. O průběhu procesu informuje grafické znázornění průběhu, které je umístěno v horní části obrazovky, a také informace o čase průběhu, které jsou umístěny v části *Výsledky*.

Po dokončení identifikace jsou zobrazeny výsledky celého procesu v pravé části obrazovky, viz obr. 9.2. Je také možné prohlédnout si výsledky předzpracování, kliknutím na tlačítko *Histogramy*. Dále je možné zobrazit výsledky segmentace, pomocí tlačítka *Barevná segmentace* nebo tabulky klasifikace, stisknutím tlačítka *Detekce tvaru*. V každém kroku uživatele je možné kliknout na ikonu *i*, která vypíše vždy zprávu o možných krocích uživatele.

Pro další použití je možné otevřít další snímek, nebo pouze změnit nastavení jednotlivých kroků algoritmu a program opětovně spustit, nicméně je doporučeno nejprve stisknout tlačítko *RESET* a poté celou akci opakovat od začátku.

ZÁVĚR

Tato práce se zabývala detekcí a rozpoznáním svislého dopravního značení v obraze. Obrazy byly pořízeny vlastním fotoaparátem a byla vytvořena vlastní databáze obsahující mnoho objektů určených k detekci a identifikaci.

V první části této práce proběhlo seznámení se s teorií, za účelem následného využití k sestavení algoritmu vykonávajícího proces detekce a rozpoznání. Kvůli robustnosti navrženého algoritmu je teoretická část poměrně obsáhlá, avšak obsahuje popis prakticky všech problematik, se kterými bylo nutné se při jeho návrhu vypořádat.

Ve druhé části této práce byly využity teoretické znalosti z první části k sestavení algoritmu navrženého v programovacím prostředí Matlab. Program Matlab byl vybrán k návrhu algoritmu hned z několika důvodů. Mezi ně patří například perfektně zpracovaná dokumentace prakticky ke všem funkcím, které lze v tomto programu využít a dále pak snadná tvorba uživatelského prostředí pomocí GUIDE.

Prvotní myšlenkou bylo prosté detekování a určení značení pomocí informací získaných v barevném prostoru RGB s využitím definovaných vlastností dopravního značení. Byla získána řada informací o jednotlivých značkách a jejich barevných vlastnostech, které byly implementovány do algoritmu. Následně byl vytvořen algoritmus pro rozpoznání tvaru dopravního značení na základě poměru jeho obsahu k obsahu čtyřúhelníkové oblasti, ve které se objekt nachází. Po konzultaci takto řešeného problému bylo učiněno rozhodnutí o využití informací i z jiných barevných prostorů (HSV nebo YCbCr), které jsou pro danou úlohu v mnoha ohledech vhodnější.

Konečná podoba algoritmu je tak složena celkem ze čtyř kroků, mezi něž patří předzpracování v podobě jasové transformace, barevná segmentace, provedená prahováním, klasifikace objektů a jejich finální identifikace. Celý algoritmus je značně závislý na vytvořené databázi značek. Následně bylo v rámci uživatelského prostředí provedeno sestavení komplexního algoritmu, který umožňuje výběr z více metod předzpracování, barevných prostorů k segmentaci, metod pro klasifikaci objektů a rozpoznávacích funkcí. Do programu byla implementována další metoda detekce objektu, a sice detekce pomocí Houghovy transformace. Dále byly sestaveny dvě metody finální identifikace značení, první založená na metodě SURF a druhá metoda založená na principu template matchingu.

Výsledný algoritmus má nejlepší úspěšnost bez použití předzpracování, se segmentací v prostoru HSV, detekcí objektů založenou na regionech a identifikací pomocí metody template matchingu. Úspěšnost této metody na testovaných dvaceti

značkách je 85%. Celková úspěšnost, testovaná na všech snímcích, pak činí pouze 49%, většina snímků však svou nedostačující kvalitou značně snižuje celkovou úspěšnost.

Vzhledem k sestavení bílého podobrazu je možné rozšířit program o identifikaci dodatkových značek nebo vodorovného dopravního značení. Dále je možné doplnit databáze značek o další vzory. A také doplnit algoritmus o zelený podobraz, který by spolu s možnou sestavenou funkcí pro rozpoznání textu mohl identifikovat více svislých dopravních značek, jako například informační značky všeho druhu a značky dodatkové, obsahující nápisy.

LITERATURA

- [1] DVOŘÁK, M. *Detekce a rozpoznání dopravního značení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 62s. Vedoucí diplomové práce Ing. Petr Honec, Ph.D.
- [2] ZÁMEČNÍK, D. a VESELÝ, I. *Rozpoznávanie červených dopravných značiek s použitím Houghovej transformácie a neurónových sietí*. Elektro revue. 2010, 2010(3), 1-7. ISSN 1213- 1539.
- [3] JUN, WooYeol, JeongMok HA, Byeongchan JEON, JoonHo LEE a Hong JE-ONG. *LED traffic sign detection with the fast radial symmetric transform and symmetric shape detection*. 2015 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2015, 310-315. DOI: 10.1109/IVS.2015.7225704. ISBN 978-1-4673-7266-4. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7225704>
- [4] MUDROVÁ, M. *Barvy v počítačové grafice* [online prezentace]. Praha: Ústav počítačové a řídicí techniky ,VŠCHT, [cit. 2015-10-3]. Dostupné z: <http://uprt.vscht.cz/mudrova/zob/prednasky/02-BARVY/Barvy.pdf>
- [5] PIHAN, R. *Gamut*. SLOVNÍK FOTOGRAFICKÝCH POJMŮ; [online]. © 2011 [cit. 2015-10-15]. Dostupné z: http://fotoroman.cz/glossary2/3_gamut.htm
- [6] MARQUES, O. *Practical image and video processing using MATLAB.*, Hoboken, N.J.: Wiley-IEEE Press, 2011, 1, 639 p. ISBN 978-0-470-04815-3.
- [7] ŽÁRA, J. *Moderní počítačová grafika. 2.*, přeprac. a rozš. vyd. Praha: Computer Press, 2004, s. 542-546. ISBN 80-251-0454-0.
- [8] KOMPRDOVÁ, K. *Rozhodovací stromy a lesy*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2012, 98 s. ISBN 978-80-7204-785-7.
- [9] *Algoritmus ID3* [online]. 2002 [cit. 2015-11-27]. Dostupné z: <http://datamining.xf.cz/view.php?cislocclanku=2002102803>
- [10] HORÁK, K. *Jasové transformace* [online prezentace]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizační techniky, [cit. 2015-12-11]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/04_Jasove_transformace.pdf
- [11] HLAVÁČ, V. *PŘEDZPRACOVÁNÍ v prostoru obrazů* [online prezentace]. Praha: Fakulta elektrotechnická, Katedra kybernetiky, Centrum strojího

- vnímání, ČVUT, [cit. 2015-12-11]. Dostupné z: <http://cmp.felk.cvut.cz/~hlavac/Public/TeachingLectures/Predzprac0br.pdf>
- [12] BOLEČEK, L. *Zobrazování černobílých snímků v nepravých barvách*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. s. 21-25. Vedoucí diplomové práce prof. Ing. Václav Říčný, CSc..
- [13] PAVELEK, M., JANOTKOVÁ, E. a ŠTĚTINA, J. *Vizualizační a optické měřicí metody* [online skriptum]. © 2001, 2007 [cit. 2016-4-15]. Dostupné z: <http://ottp.fme.vutbr.cz/~pavelek/optika/1508.htm>
- [14] HORÁK, K. *Detekce Hran a rohů* [online prezentace]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizační techniky, [cit. 2015-12-11]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/07_Detekce_hran_a_rohu.pdf
- [15] HLAVÁČ, V. *Hledání hran* [online prezentace]. Praha: Fakulta elektrotechnická, katedra kybernetiky, Centrum strojového vnímání, ČVUT, [cit. 2015-12-11]. Dostupné z: <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/22EdgeDetectionCz.pdf>
- [16] CANNY, John. *A Computational Approach to Edge Detection* [online]. 6.11.1986, s. 679-698 [cit. 2015-12-11]. Dostupné z: <http://cmp.felk.cvut.cz/~cernyad2/TextCaptchaPdf/A%20Computational%20Approach%20to%20Edge%20Detection.pdf>
- [17] KALOVÁ, I. *Segmentace* [online prezentace]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizační techniky, [cit. 2015-12-11]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/Lectures/04_Segmentace.pdf
- [18] HLAVÁČ, V. *Matematická morfologie* [online prezentace]. Praha: Fakulta elektrotechnická, katedra kybernetiky, Centrum strojového vnímání, ČVUT, [cit. 2016-4-16]. Dostupné z: <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/71-3MatMorpholBinCz.pdf>
- [19] HORÁK, K. *Matematická morfologie* [online prezentace]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizační techniky, [cit. 2016-4-16]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/10_Matematicka_morfologie.pdf
- [20] *Dokumentace softwaru Matlab* [online]. [cit 2015-11-21] Dostupné z: <http://www.mathworks.com/help/images/>

- [21] HORÁK, K. *Popis objektů* [online prezentace]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizační techniky, [cit. 2016-4-16]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/Lectures/07_Popis_objektu.pdf
- [22] THURLEY ,M. a CARLSON, J. *The Hough Transform* [online prezentace]. 2010 [cit. 2016-4-15]. Dostupné z: http://www.ltu.se/cms_fs/1.36192!/e0005e_lecture05_hough_transform.dvi.pdf
- [23] MACDONALD, I. *Probabilistic Hough Transform* [online]. 2011 [cit. 2016-4-15]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/macdonald.pdf
- [24] HRÚZ, M. *SIFT, SURF, MSER* [online]. 29.10.2015. Plzeň: Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra kybernetiky, [cit. 2016-4-29]. Dostupné z: <http://www.kky.zcu.cz/uploads/courses/mpv/04/materialy04.pdf>
- [25] *Vyhláška Ministerstva dopravy a spojů, kterou se provádějí pravidla provozu na pozemních komunikacích a úprava a řízení provozu na pozemních komunikacích*. In: 30/2001 Sb. 2001. s. 541-558. Dostupné z: <http://aplikace.mvcr.cz/sbirkazakonu/ViewFile.aspx?type=c&id=3589>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

RGB - *Red, Green, Blue*

CMY - *Cyan, Magenta, Yellow*

HSV - *Hue, Saturation, Value*

YCbCr - Y značí jasový kanál, Cb a Cr označují modrý a červený chrominanční komponent

LUT - *LookUp Table* vyhledávací tabulka

MM - Matematická Morfologie

SIFT - *Scale-Invariant Feature Transform*

SURF - *Speeded-Up Robust Features*

GUIDE - *Graphical User Interface Development Environment*

$Q(T)$ minimum kvadratické chyby

GI gini index

H entropie

ME klasifikační chyba

$GAIN$ informační zisk

A OBSAH PŘILOŽENÉHO CD

- Elektronická verze bakalářské práce,
- zdrojové soubory programu,
- databáze vzorových značek,
- pořízené snímky.